

ANALISIS PERBANDINGAN ALGORITME K-MEANS DAN ISODATA UNTUK KLASTERISASI DATA KEJADIAN TITIK API DI WILAYAH SUMATERA PADA TAHUN 2001 HINGGA 2014

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Edo Fadila Sirat
NIM: 145150209111017



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

ANALISIS PERBANDINGAN ALGORITME K-MEANS DAN ISODATA UNTUK
KLASTERISASI DATA KEJADIAN TITIK API DI WILAYAH SUMATERA PADA TAHUN
2001 HINGGA 2014

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Edo Fadila Sirat
NIM: 145150209111017

Skripsi ini telah diuji dan dinyatakan lulus pada
6 Juni 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Budi Darma Setiawan, S.Kom, M.Cs
NIP: 198410152014041002

D.Sc Fatwa Ramdani, S.Si, M.Sc.
NIK: 2016118506191001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 197105182003121001

IDENTITAS TIM PENGUJI

Penguji 1:

Putra Pandu Adikara S.Kom, M.Kom

NIP. 19850725 200812 1 002

Penguji 2:

Dr.Eng. Fitra Abdurrachman Bachtiar, S.T, M.Eng

NIK 201201 840628 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 16 April 2018

Edo Fadila Sirat

NIM: 145150209111017



DAFTAR RIWAYAT HIDUP

Nama Lengkap : Edo Fadila Sirat
 Tempat, tanggal lahir : Pamekasan, 20 Maret 1993
 Jenis kelamin : Laki laki
 Agama : Islam
 Kewarganegaraan : Indonesia
 Alamat : Jl Masjid Patemon 43 RT 02 RW 02 Kab. Pamekasan Jawa Timur
 Riwayat Pendidikan : - TK Al Munawarah Pamekasan tahun 1997-1999
 - SDN Patemon 1 Pamekasan tahun 1999-2005
 - SMPN 1 Pamekasan tahun 2005-2008
 - SMAN 1 Pamekasan tahun 2008-2011
 - Universitas Brawijaya Program Studi Manajemen Informatika dan Teknik Komputer 2011-2014

Malang, Juli 2018

Penulis

Edo Fadila Sirat

UCAPAN TERIMA KASIH



ABSTRAK

Edo Fadila Sirat, Analisis Perbandingan Algoritme K-Means dan Isodata untuk Klasterisasi Data Kejadian Titik Api di Wilayah Sumatera pada Tahun 2001 hingga 2014

Pembimbing : Budi Darma Setiawan, S.Kom, M.Cs. dan D.Sc Fatwa Ramdani, S.Si, M.Sc.

Fenomena Kebakaran merupakan fenomena yang tidak asing di Indonesia. Tingginya angka kejadian kebakaran yang terjadi di Indonesia membutuhkan perhatian khusus dari pemerintah, agar setiap bencana alam seperti kebakaran hutan dapat ditanggulangi. Hasil pantauan satelit tercatat pada sebuah file data titik api dengan ukuran data yang cukup besar sehingga data sulit diolah untuk menjadi informasi yang mudah terima oleh pengguna. Berdasarkan data yang diperoleh dari situs EOSDIS tercatat sebanyak 289.256 kejadian titik api dalam rentan waktu antara 2001 hingga 2014. Dibutuhkan sebuah algoritme untuk melakukan segmentasi data atau klasterisasi data, agar data yang besar dapat diolah menjadi sebuah informasi yang baik bagi pengguna. Dalam penelitian ini dilakukan studi perbandingan algoritme klasterisasi antara *K-Means* dan *Isodata*. Kedua algoritme yang digunakan dalam penelitian ini dinilai berdasarkan kualitas kluster yang dihasilkan. Algoritme yang digunakan dalam mengukur kualitas kluster dalam penelitian ini adalah *Silhouette Coefficient* (SC). Hasil akhir nilai SC algoritme *K-Means* sebesar 0.999997187 dan algoritme *Isodata* sebesar 0.999957161, sehingga dalam hal ini, algoritme *K-Means* memiliki nilai SC yang lebih tinggi dibandingkan dengan algoritme *Isodata* dalam mengkluster data kejadian titik api dengan selisih nilai SC yang kecil.

Kata kunci: k-means, Isodata, klasterisasi, Kejadian titik api, titik panas

ABSTRACT

Edo Fadila Sirat, Comparative Analysis of K-Means and Isodata Algorithms for Clustering of Fire Point Data in Sumatra Region from 2001 to 2014

**Mentor : Budi Darma Setiawan, S.Kom, M.Cs. dan D.Sc Fatwa Ramdani, S.Si,
M.Sc**

Fire phenomenon is a familiar phenomenon in Indonesia. The high number of fire incidents in Indonesia requires special attention from the government, so that any natural disasters such as forest fires can be overcome. Satellite monitoring results are recorded on a data file of fire points with a large enough data numbers so that the data is difficult to be processed to become information that is easily received by the user. Based on data obtained from the EOSDIS site recorded as many as 289,256 fire spots occurrence in the timeframe between 2001 to 2014. It takes an algorithm to segment the data or cluster the data, so that large data can be processed into a good information for the user. In this study a comparative study of clustering algorithms between K-Means and Isodata was conducted. Both algorithms used in this study were assessed based on the quality of the clusters produced. The algorithm used in measuring the quality of cluster in this research is Silhouette Coefficient (SC). The final result value of Silhouette Coefficient K-Means method is 0.999997187 and Isodata method is 0.999957161, so in this case, K-Means algorithm has a higher SC value compared to the Isodata algorithm in clustering the data of fire spots with a small SC value difference.

Keywords: *k-means, Isodata, clustering, Fire point events, hot spot*

KATA PENGANTAR

Puji Syukur Kehadirat Allah SWT yang telah melimpahkan rahmat dan memberikan kemudahan kepada penulis sehingga dapat menempuh skripsi dengan judul “ Analisis perbandingan algoritme *K-Means* dan *Isodata* untuk Klasterisasi Kejadian Titik Api Wilayah Sumatera pada Tahun 2001 hingga 2014” Tujuan disusunnya laporan skripsi ini untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer.

Dalam skripsi ini metode yang dipilih adalah *K-Means* dan *Isodata clustering*. *K-Means* adalah salah satu algoritme klasterisasi yang banyak digunakan dalam sebuah penelitian dibidang komputasi cerdas. Algoritme *K-Means* merupakan algoritme yang cukup simpel dalam melakukan klasterisasi data, sehingga banyak orang yang tertarik untuk menggunakan algoritme *K-Means* sebagai metode penelitian untuk klasterisasi data. Dalam hal ini, penulis tertarik untuk lebih mengeksplorasi lagi algoritme *K-Means* dalam melakukan klasterisasi data. Disisi lain penulis juga menggunakan algoritme *Isodata* dalam penelitian ini.

Isodata merupakan salah satu algoritme yang juga dapat digunakan dalam melakukan klasterisasi data. Berbeda halnya dengan *K-Means*, algoritme *Isodata* merupakan algoritme yang jarang digunakan dalam sebuah penelitian, lebih khususnya di Indonesia. Algoritme *Isodata* memiliki tahapan yang lebih kompleks dibandingkan algoritme *K-Means* dalam melakukan klasterisasi data. Dengan segala macam perbedaan yang ada pada kedua metode tersebut, penulis tertarik untuk melakukan studi perbandingan tentang *K-Means* dan *Isodata* dalam melakukan klasterisasi data. Data yang digunakan dalam skripsi ini adalah data kejadian titik api wilayah Sumatera tahun 2001 hingga 2014.

Dalam penulisan skripsi ini tak lepas berbagai macam hambatan dan kesulitan, namun berkat pertolongan Allah SWT, skripsi ini dapat di selesaikan. Penulis mengucapkan terima kasih kepada:

1. Bapak Budi Darma Setiawan, S.kom, M.Cs dan Bapak D.Sc Fatwa Ramdani, S.Si, M.Sc selaku Dosen pembimbing skripsi.
2. Bapak Bayu Priyambadha, S.kom, M.Kom selaku Ketua Program Studi Teknik Informatika.
3. Bapak Tri Astoto Kurniawan, S.T., M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
4. Ayahanda dan Ibunda yang telah memberikan dukungan doa, moriil dan materil kepada penulis.
5. Seluruh Civitas Akademik Jurusan Teknik Informatika yang telah memberikan layanan dengan sebaik baiknya.
6. Teman teman Mahasiswa Fakultas Ilmu Komputer yang telah memberikan dukungan doa maupun bantuan dalam pengerjaan skripsi ini.

7. Pihak pihak lain yang telah membantu namun tidak dapat disebut satu persatu.

Semoga dukungan maupun bantuan yang diberikan kepada penulis oleh pihak pihak yang terkait mendapatkan balasan dari Allah SWT. Penulis menyadari dalam penyusunan laporan skripsi ini masih jauh dari kata sempurna, sehingga penulis mengharapkan kritik dan saran yang membangun dari pembaca. Semoga dengan adanya penulisan skripsi ini dapat memberikan manfaat bagi pembaca.

Malang, 16 April 2018

Penulis

edofadila@gmail.com



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iv
KATA PENGANTAR.....	vii
ABSTRAK.....	vii
ABSTRACT	viii
DAFTAR ISI	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 <i>The Earth Observation System Data and Information System (EOSDIS)</i>	9
2.2 Min max normalisasi	10
2.3 Algoritme <i>K-Means</i>	10
2.4 Algoritme <i>Isodata</i>	11
2.5 <i>Silhouette Coefficient</i>	13
BAB 3 METODOLOGI DAN PERANCANGAN.....	14
3.1 Studi pustaka	14
3.2 Pengumpulan data.....	14
3.3 Analisis kebutuhan.....	16
3.3.1 Perangkat keras.....	17
3.3.2 Perangkat lunak.....	17
3.4 Perancangan sistem	17
3.4.1 Algoritme <i>K-Means</i>	17

3.4.2 Manualisasi algortime <i>K-Means</i>	19
3.4.3 Algoritme <i>Isodata</i>	30
3.4.4 Manualisasi Algoritme <i>Isodata</i>	32
3.5 Implementasi	43
3.6 Pengujian dan Analisis	43
BAB 4 IMPLEMENTASI	44
4.1 Implementasi antarmuka.....	44
4.1.1 Antarmuka program <i>K-Means</i>	44
4.1.2 Antarmuka awal <i>K-Means</i>	44
4.1.3 Antarmuka input variabel <i>K-Means</i>	45
4.1.4 Antarmuka hasil <i>scrool</i> atas <i>K-Means</i>	46
4.1.5 Antarmuka hasil <i>scroll</i> bawah <i>K-Means</i>	47
4.1.6 Antarmuka program <i>Isodata</i>	48
4.1.7 Antarmuka awal <i>Isodata</i>	48
4.1.8 Antarmuka masukan variabel <i>Isodata</i>	49
4.1.9 Antarmuka hasil <i>scroll</i> atas <i>Isodata</i>	50
4.1.10 Antarmuka hasil <i>scroll</i> bawah <i>Isodata</i>	51
4.2 Implementasi algoritme.....	52
4.2.1 Implementasi algoritme <i>K-Means</i>	52
4.2.2 Implementasi algoritme <i>Isodata</i>	58
BAB 5 PENGUJIAN DAN ANALISIS.....	65
5.1 Uji coba <i>K-Means</i>	66
5.1.1 Uji coba <i>K</i>	66
5.1.2 Uji coba <i>Threshold</i>	67
5.2 Uji coba <i>Isodata</i>	68
5.2.1 Uji coba <i>K₀</i>	68
5.2.2 Uji coba <i>n_{min}</i>	69
5.2.3 Uji coba <i>σ²_{max}</i>	70
5.2.4 Uji coba <i>d_{min}</i>	71
5.2.5 Uji coba <i>Iterasi Max</i>	72
5.3 Analisis perbandingan.....	73
5.3.1 Persamaan.....	74

5.3.2 Perbedaan	74
5.3.3 Perbandingan kualitas klaster	74
BAB 6 penutup	76
6.1 Kesimpulan.....	76
6.2 Saran	76
DAFTAR PUSTAKA.....	77
LAMPIRAN A MANUALISASI SILHOUETTE COEFFICIENT	79
LAMPIRAN B UJI COBA K-MEANS.....	81
LAMPIRAN C UJI COBA ISODATA.....	83



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	7
Tabel 3.1 Rincian kebutuhan Perangkat Keras.....	17
Tabel 3.2 Rincian pendukung perangkat lunak.....	17
Tabel 3.3 Data sampel kejadian titik api	19
Tabel 3.4 Normalisasi data sampel <i>K-Means</i>	20
Tabel 3.5 Alokasi data pada <i>KK-Means</i>	20
Tabel 3.6 Alokasi data klaster 1 (<i>K-Means</i>).....	21
Tabel 3.7 Alokasi data klaster 2 (<i>K-Means</i>).....	21
Tabel 3.8 Alokasi data klaster 3 (<i>K-Means</i>).....	22
Tabel 3.9 Pusat klaster (<i>K-Means</i>)	22
Tabel 3.10 Jarak data ke <i>Centroid</i> (<i>K-Means</i>)	23
Tabel 3.11 Alokasi data pada tiap K iterasi 1 (<i>K-Means</i>).....	24
Tabel 3.12 Alokasi data klaster 1 Iterasi 1 (<i>K-Means</i>).....	24
Tabel 3.13 Alokasi data klaster 2 iterasi 1 (<i>K-Means</i>).....	25
Tabel 3.14 Alokasi data klaster 3 iterasi 1 (<i>K-Means</i>).....	25
Tabel 3.15 Pusat klaster iterasi 1 (<i>K-Means</i>).....	26
Tabel 3.16 Jarak data ke <i>Centroid</i> iterasi 1 <i>K-Means</i>	26
Tabel 3.17 Alokasi data pada tiap K iterasi 2 <i>K-Means</i>	27
Tabel 3.18 Alokasi data klaster 1 iterasi 2 <i>K-Means</i>	27
Tabel 3.19 Alokasi data klaster 2 iterasi 2 <i>K-Means</i>	28
Tabel 3.20 Alokasi data klaster 3 iterasi 2 <i>K-Means</i>	28
Tabel 3.21 Pusat klaster iterasi 2 <i>K-Means</i>	29
Tabel 3.22 Jarak data ke <i>Centroid</i> iterasi 2 <i>K-Means</i>	29
Tabel 3.23 Data sampel <i>Isodata</i>	32
Tabel 3.24 Normalisasi data sampel <i>Isodata</i>	33
Tabel 3.25 Alokasi data pada <i>K Isodata</i>	33
Tabel 3.26 Alokasi data klaster 1 <i>Isodata</i>	34
Tabel 3.27 Alokasi data Klaster 2 <i>Isodata</i>	34
Tabel 3.28 Alokasi data klaster 3 <i>Isodata</i>	35
Tabel 3.29 <i>Centroid Isodata</i>	35

Tabel 3.30 Jarak data Ke <i>Centroid Isodata</i>	36
Tabel 3.31 Status klaster <i>Isodata</i>	36
Tabel 3.32 Pembaruan klaster <i>Isodata</i>	36
Tabel 3.33 Pembaruan data klaster 1 <i>Isodata</i>	37
Tabel 3.34 Pembaruan data klaster 3 <i>Isodata</i>	37
Tabel 3.35 Pembaruan pusat klaster (<i>Isodata</i>)	38
Tabel 3.36 Jarak Data Ke <i>Centroid</i> Iterasi 1 <i>Isodata</i>	38
Tabel 3.37 Status klaster iterasi 1 <i>Isodata</i>	39
Tabel 3.38 Pembaruan Klaster Iterasi 1 <i>Isodata</i>	39
Tabel 3.39 Pembaruan data klaster 1 <i>Isodata</i>	39
Tabel 3.40 Pembaruan data klaster 3 iterasi 1 <i>Isodata</i>	40
Tabel 3.41 Pembaharuan Pusat Klaster Iterasi 1 (<i>Isodata</i>)	40
Tabel 3.42 Jarak data ke centroid iterasi 2 <i>Isodata</i>	41
Tabel 3.43 Pembaruan klaster iterasi 2 <i>Isodata</i>	41
Tabel 3.44 Pembaruan klaster 1 iterasi 2 <i>Isodata</i>	42
Tabel 3.45 Pambaruan klaster 3 iterasi 2 <i>Isodata</i>	42
Tabel 3.46 Pembaruan Pusat Klaster Iterasi 2 <i>Isodata</i>	43
Tabel 4.1 Implementasi proses <i>Split Isodata</i>	61
Tabel 5.1 Uji coba variabel K	66
Tabel 5.2 uji coba varabel <i>Threshold</i>	67
Tabel 5.3 Uji coba K_0	68
Tabel 5.4 Uji coba n_{min}	70
Tabel 5.5 Uji coba σ^2_{max}	71
Tabel 5.6 Uji coba d_{min}	72
Tabel 5.7 Uji coba Iterasi Max	73
Tabel 5.8 perbandingan nilai SC	74

DAFTAR GAMBAR

Gambar 3.1 Diagram Metodologi	14
Gambar 3.2 Peta Seleksi Wilayah.....	15
Gambar 3.3 Data Kejadian Titik Api	15
Gambar 3.4 Diagram alir Algoritme <i>K-Means</i>	18
Gambar 3.5 Diagram alir Algoritme <i>Isodata</i>	31
Gambar 4.1 Alur implementasi	44
Gambar 4.2 Tampilan awal program <i>K-Means</i>	45
Gambar 4.3 Antarmuka masukan variabel <i>K-Means</i>	46
Gambar 4.4 Antarmuka hasil <i>scroll</i> atas <i>K-Means</i>	47
Gambar 4.5 Antarmuka hasil <i>scroll</i> bawah <i>K-Means</i>	48
Gambar 4.6 Antarmuka Tampilan Awal <i>Isodata</i>	49
Gambar 4.7 Antarmuka masukan variabel <i>Isodata</i>	50
Gambar 4.8 Antarmuka Hasil Scroll Atas <i>Isodata</i>	51
Gambar 4.9 Antarmuka Scroll Bawah <i>Isodata</i>	52
Gambar 5.1 Uji Coba	65
Gambar 5.2 Grafik pengaruh k terhadap SC	67
Gambar 5.3 Grafik pengaruh <i>threshold</i> terhadap SC.....	68
Gambar 5.4 Grafik pengaruh K_0 terhadap SC.....	69
Gambar 5.5 Grafik pengaruh n_{min} terhadap SC.....	70
Gambar 5.6 Grafik pengaruh σ^2_{max} terhadap SC	71
Gambar 5.7 Grafik pengaruh d_{min} terhadap SC	72
Gambar 5.8 Grafik pengaruh <i>Iterasi max</i> terhadap SC.....	73

DAFTAR LAMPIRAN

LAMPIRAN A MANUALISASI SILHOUETTE COEFFICIENT	79
LAMPIRAN B UJI COBA K-MEANS.....	81
B.1 Uji coba variabel K	81
B.2 Grafik uji coba variabel K	81
B.3 Uji coba variabel <i>Threshold</i>	81
B.4 Grafik Uji coba Variabel <i>threshold</i>	82
LAMPIRAN C UJI COBA ISODATA.....	83
C.1 Uji coba K_0	83
C.2 Uji coba n_{min}	83
C.3 Uji Coba variabel Varian max (σ^2_{max})	84
C.4 Uji coba variabel d_{min}	85
C.5 Uji coba Iterasi max	85



BAB 1 PENDAHULUAN

1.1 Latar belakang

Fenomena Kebakaran merupakan fenomena yang tidak asing di Indonesia. Maraknya kejadian seperti kebakaran hutan menambah daftar panjang peristiwa kejadian kebakaran di beberapa wilayah Indonesia. Terdapat beberapa faktor yang menyebabkan terjadinya peristiwa kebakaran hutan. Peristiwa kebakaran dapat terjadi dikarenakan faktor alami maupun faktor perbuatan manusia (Rasyid, 2014).

Pada umumnya faktor alami yang menjadi pemicu kejadian peristiwa kebakaran adalah cuaca terik pada musim kemarau hingga terjadi kekeringan yang berdampak pada lahan gambut (Harrison, et al., 2009). Secara umum kekeringan yang terjadi di Indonesia dikarenakan iklim di Indonesia sangat dipengaruhi oleh El Nino. Kebakaran yang terjadi Indonesia, banyak membakar lahan gambut yang mengering dan mudah terbakar. Lahan gambut yang terbakar seringkali terjadi disebabkan oleh perbuatan manusia (Harrison, et al., 2009).

Faktor yang menyebabkan terjadinya kebakaran yaitu adanya alih fungsi lahan yang dilakukan oleh manusia. Adanya perusahaan yang ingin mengambil sebagian lahan di sekitar hutan untuk alokasi perkebunan dengan cara melakukan pembakaran hutan, dapat memicu terjadinya kebakaran hutan yang sengaja dilakukan oleh manusia. Begitu halnya ketika masyarakat sekitar ingin membuka lahan pertanian dengan melakukan alih fungsi lahan hutan. Alih fungsi lahan tersebut dilakukan dengan cara pembakaran hutan. Faktor lain yang disebabkan oleh kelalaian manusia seperti membuang puntung rokok sembarangan disekitar lahan kering, dapat memicu terjadinya kebakaran (Harrison, et al., 2009).

Titik api atau yang lebih dikenal dengan istilah titik panas (*hotspot*) merupakan indikator terjadinya kebakaran hutan yang mendeteksi suatu lokasi memiliki suhu lebih tinggi dibandingkan dengan suhu lokasi lainnya. Titik panas adalah hasil deteksi kebakaran hutan/lahan pada ukuran piksel tertentu (1km x 1km) yang memungkinkan terjadinya kebakaran pada saat satelit melintas pada kondisi relative bebas awan menggunakan algoritme tertentu (Giglio, et al., 2003). Selang kepercayaan atau *convidence* menjadi indikator untuk mengukur tingkat keyakinan terjadinya kebakaran. Semakin tinggi nilai *convidence* maka semakin tinggi tingkat keyakinan bahwa adanya kebakaran hutan di suatu lokasi (Parwati, et al., 2016).

Salah satu wilayah di Indonesia yang rentan mengalami kejadian titik api adalah pulau Sumatera. Berdasarkan data yang diperoleh dari situs EOSDIS tercatat sebanyak 289.256 kejadian titik api dalam rentan waktu antara 2001 hingga 2014. Tingginya angka kejadian kebakaran yang terjadi di pulau Sumatera membutuhkan perhatian khusus dari pemerintah maupun masyarakat setempat, agar setiap bencana alam seperti kebakaran hutan maupun bencana alam lainnya dapat ditanggulangi. Terjadinya peristiwa kebakaran dapat dipantau melalui satelit. Hasil pantauan satelit tercatat pada sebuah file data titik api dengan ukuran

data yang cukup besar sehingga data sulit diolah untuk menjadi informasi yang mudah diterima oleh pengguna. Diperlukan sebuah metode untuk melakukan segmentasi data agar data yang besar dapat diolah menjadi sebuah informasi yang baik bagi pengguna.

Pada penelitian sebelumnya yang dilakukan oleh Pramesty dkk, membahas tentang implementasi *K-Medoids Clustering* untuk pengelompokan data potensi kebakaran hutan/lahan berdasarkan titik panas. Dalam penelitian ini metode *K-Medoid* digunakan untuk mengkluster data titik panas atau titik api. Parameter yang digunakan dalam penelitian ini adalah *latitude*, *longitude*, *brightness*, *frp* dan *confidence*. Hasil kluster dalam penelitian ini, diuji dengan metode *Shilhouette Coefficient* (SC). Nilai SC terbaik yang didapat dalam penelitian ini sebesar 0.5745 pada penggunaan 2 kluster untuk 7352 data (Pramesti, et al., 2017). Adanya beberapa metode lain yang dapat digunakan untuk melakukan segmentasi data diantaranya adalah Metode *K-Means* dan *Isodata*. Kedua metode tersebut merupakan metode klusterisasi yang mendapatkan hasil uji yang cukup baik berdasarkan pengamatan dari penelitian-penelitian sebelumnya.

Pada penelitian yang dilakukan oleh Muningsih dan Kiswati (2015) dengan membahas tentang pengelompokan produk yang dijual pada *online shop* ragam Jogja menjadi beberapa kluster untuk mengetahui produk mana yang paling banyak diminati sehingga dapat dilakukan penyesuaian dengan stok. Variabel yang digunakan adalah kode produk, jumlah transaksi, volume penjualan dan rata-rata penjualan. Pengukuran dan evaluasi dalam penelitian ini menghasilkan 3 kluster dengan tingkat akurasi tinggi pada kluster 1 mencapai 80%, kluster 2 mencapai 60%, dan kluster 3 mencapai 80% (Muningsih & Kiswati, 2015).

Pada penelitian yang dilakukan oleh Ghosh dan Dubey (2013), membahas tentang analisis perbandingan dua algoritme yaitu *K-Means* dan *Fuzzy C-Means* dalam melakukan analisis data. Penelitian ini diuji berdasarkan kompleksitas waktu komputasi antara dua algoritma. Algoritme *K-Means* menghasilkan waktu 0.443755 detik sedangkan pada algoritme FCM menghasilkan waktu 0.781679 detik (Ghosh & Dubey, 2013).

Pada penelitian yang dilakukan oleh Yu dkk, membahas tentang algoritme *Fuzzy Isodata* untuk penilaian resiko kredit dari *Emerging Information Technology* (EIT). Dalam penelitian ini terdapat beberapa indikator sebagai parameter penilaian. Indikator penilaian dalam penelitian ini terdiri dari 4 indikator utama dan 15 indikator sekunder. Empat indikator utama yaitu ilmiah, operasional, prinsip objektif serta kombinasi kuantitatif dan indeks evaluasi. Metode yang digunakan dalam penelitian ini adalah *Fuzzy Isodata*. Hasil yang diperoleh dari klusterisasi algoritma *Isodata* menghasilkan 10 investigasi perusahaan EIT harus ditetapkan menjadi 5 kredit nilai sehingga dalam penelitian ini algoritme *Isodata* merupakan hal yang tepat apabila diterapkan pada perusahaan EIT (Yu, et al., 2012).

Pada penelitian yang dilakukan oleh Liu dkk (2012), membahas tentang pemilihan fitur dari data ekspresi gen berdimensi tinggi. Dalam penelitian ini data yang digunakan adalah data dalam bidang bioinformatika. Dalam penelitian ini,

metode yang diusulkan adalah algoritma *Fuzzy Isodata*. Evaluasi hasil dalam penelitian ini, dilakukan dengan 5 *microarray dataset* diantaranya, *DLBCL*, *Acute Leukemia*, *Multiple myeloma*, *Colon* dan *Prostate*. Hasil yang diperoleh menunjukkan bahwa fitur yang dipilih memiliki kinerja yang baik dalam melakukan klasifikasi dan klusterisasi pada 5 *gen dataset* profil ekspresi, sehingga dalam penelitian ini disimpulkan bahwa percobaan yang dilakukan dengan *isodata RFE* mampu dalam memilih fitur dengan klasifikasi yang kuat dan kemampuan klusterisasi (Liu, et al., 2012).

Berdasarkan beberapa penelitian sebelumnya, metode *K-Means* dan *Isodata* dapat diterapkan pada berbagai macam studi kasus dengan cukup baik. Pada metode *K-Means* mendapatkan penilaian yang cukup apabila ditinjau dari segi kualitas kluster dan waktu komputasi, sedangkan metode *isodata* mendapatkan nilai yang cukup baik apabila ditinjau dari segi kualitas kluster. Dalam penelitian ini kedua metode tersebut akan diterapkan untuk klusterisasi data kejadian titik api wilayah Sumatera pada tahun 2001 hingga 2014.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijabarkan, dapat ditarik beberapa rumusan masalah sebagai berikut:

1. Bagaimana kualitas kluster algoritme *K-Means* pada studi kasus klusterisasi data titik api di wilayah Sumatera pada tahun 2001 hingga 2014?
2. Bagaimana kualitas kluster algoritme *Isodata* pada studi kasus klusterisasi data titik api di wilayah Sumatera pada tahun 2001 hingga 2014?
3. Bagaimana analisis perbandingan algoritme *K-Means* dan *Isodata* pada studi kasus klusterisasi data kejadian titik api di wilayah Sumatera pada tahun 2001 hingga 2014?

1.3 Tujuan

Adapun tujuan dari penelitian ini antara lain:

1. Dapat mengetahui kualitas kluster algoritme *K-Means* pada studi kasus klusterisasi data kejadian titik api di wilayah Sumatera pada tahun 2001 hingga 2014.
2. Dapat mengetahui kualitas kluster algoritme *Isodata* pada studi kasus klusterisasi data kejadian titik api di wilayah Sumatera pada tahun 2001 hingga 2014.
3. Mengetahui hasil analisis perbandingan dari algoritme *K-Means* dan *Isodata* pada studi kasus klusterisasi data kejadian titik api di wilayah Sumatera pada tahun 2001 hingga 2014.

1.4 Manfaat

Adapun manfaat yang didapat dari penelitian ini antara lain:

1. Memudahkan pengguna dalam melakukan segmentasi data kejadian titik api dengan adanya sistem klasterisasi.
2. Memudahkan pengguna dalam menemukan pola data kejadian titik api yang di klaster dengan metode *K-Means* dan *Isodata*
3. Memberikan wawasan terhadap pembaca tentang hasil evaluasi studi perbandingan dua metode *K-Means* dan *Isodata*.

1.5 Batasan masalah

Adapun batasan masalah yang ditentukan dalam penelitian antara lain:

1. Dalam penelitian ini data yang digunakan adalah data kejadian titik api pada wilayah Sumatera pada tahun 2001 hingga 2014.
2. Data yang digunakan dalam penelitian ini adalah data yang diperoleh dari earthdata.nasa.gov.
3. Dalam penelitian metode yang dipilih adalah *K-Means* dan *Isodata*
4. Dalam penelitian ini bahasan terkait dengan studi perbandingan dua metode antara *K-Means* dan *Isodata*.
5. Fitur yang digunakan penelitian ini diantaranya *latitude*, *longitude*, *brightness*, *acq_date*, *convidence*.

1.6 Sistematika pembahasan

Sistematika pembahasan dalam laporan penelitian ini berisi tentang tahapan pembahasan yang dibagi menjadi tujuh bagian yaitu:

BAB 1 PENDAHULUAN

Pada bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah dan Sistematika pembahasan

BAB 2 KAJIAN PUSTAKA

Pada bab ini membahas tentang kajian pustaka serta landasan teori yang menjadi rujukan pada penelitian ini.

BAB 3 METODOLOGI DAN PERANCANGAN

Pada bab ini membahas tentang tahapan penelitian yang dilakukan oleh penulis untuk menyelesaikan permasalahan dan perancangan sistem yang akan dibangun untuk menyelesaikan permasalahan yang dibahas.

BAB 4 IMPLEMENTASI

Pada Bab ini berisi tentang implementasi sistem yang dibangun berdasarkan perancangan yang telah dijabarkan.

BAB 5 PENGUJIAN DAN ANALISIS

Pada bab ini berisi tentang evaluasi hasil dari penerapan metode untuk menyelesaikan permasalahan dalam studi kasus yang bersangkutan.

BAB 6 PENUTUP

Pada bab ini berisi tentang kesimpulan akhir dan saran yang membangun dalam penelitian ini.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan membahas tentang landasan kepustakaan yang terkait dengan penelitian ini yang membahas tentang uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah berkaitan dengan permasalahan yang diangkat.

Pada penelitian yang dilakukan oleh Muningsih dan Kiswati (2015) membahas tentang pengelompokan produk yang dijual pada online shop Ragam Jogja agar menjadi beberapa klaster sehingga dapat diketahui produk-produk yang paling diminati maupun yang kurang diminati. Variabel yang digunakan adalah kode produk, jumlah transaksi, volume penjualan dan rata rata penjualan. Pengukuran dan evaluasi dalam penelitian ini menghasilkan 3 klaster dengan tingkat akurasi tinggi pada klaster 1 mencapai 80%, klaster 2 mencapai 60%, dan klaster 3 mencapai 80 % (Muningsih & Kiswati, 2015).

Pada penelitian yang dilakukan oleh Ghosh dan Dubey (2013) membahas tentang analisa perbandingan dua algoritma yaitu *K-Means* dan *Fuzzy C-Means* dalam melakukan analisis data. Pada penelitian ini diuji berdasarkan kompleksitas waktu komputasi antara dua algoritma. Algoritme *K-Means* menghasilkan waktu 0.443755 detik, sedangkan pada algoritma FCM menghasilkan waktu 0.781679 detik (Ghosh & Dubey, 2013).

Pada penelitian yang dilakukan Rohmawati, dkk (2015) membahas tentang pengelompokan data pelamar beasiswa bantuan (BBM) menjadi 3 kelompok sehingga dapat memudahkan dalam menentukan penerima beasiswa BBM. Dalam penelitian ini dilakukan perhitungan perbandingan nilai *purity measure* pada hasil klasterisasi dari algoritme *K-Means* dengan format atribut dataset yang berbeda-beda. Nilai *purity* pada dataset kodifikasi sebagian untuk hasil klaster algoritma *K-Means* sebesar 0.611 atau 61.11%. Pada dataset kodifikasi keseluruhan nilai *purity* hasil klaster algoritme *K-Means* sebesar 0.806 atau 80.56% (Rohmawati, et al., 2015)

Pada penelitian yang dilakukan oleh Zahrotun (2015) menggunakan Metode *Clustering K-Means* dan *Agglomerative Hierarchical Clustering* (AHC) membahas tentang pengelompokan jumlah penumpang berdasarkan jalur bus *shelter* menggunakan metode *K-means* dan *AHC*. Hasil yang diperoleh dari algoritma *K-Means* adalah jumlah klaster 0 sebanyak 15 data, klaster 1 sebanyak 44 data, dan klaster 2 sebanyak 54 data. Berdasarkan hasil klaster yang didapat, dalam penelitian ini densitas atau kerapatan data masing-masing terlihat seimbang. Sehingga hal ini memperlihatkan bahwa data memiliki similaritas yang baik (Zahrotun, 2015).

Pada penelitian yang dilakukan oleh Yu, dkk (2012), terdapat beberapa indikator sebagai parameter penilaian. Indikator penilaian dalam penelitian ini terdiri dari 4 indikator utama dan 15 indikator sekunder. Empat indikator utama yaitu Ilmiah, Operasional, prinsip objektif serta kombinasi kuantitatif dan Indeks Evaluasi. Lima belas indikator sekunder dalam penelitian ini adalah pemeliharaan

asset dan indikator apresiasi, aktiva operasi indikator negara, perputaran total asset, perputaran aktiva lancar, perputaran persediaan, piutang rekening omset, *debt paying* indikator kemampuan, rasio asset kewajiban, *acidtest ratio*, *cash rasio* aliran utang, pengembangan indikator negara, tingkat pertumbuhan penjualan, tingkat akomodasi modal dan kemampuan inovasi teknologi dan aplikasi. Objek yang digunakan dalam penelitian ini di klaster dengan algoritma *Fuzzy Isodata*. Hasil yang diperoleh dari klasterisasi algortime Isodata menghasilkan 10 investigasi perusahaan *EIT* harus ditetapkan menjadi 5 kredit nilai sehingga dalam penelitian ini algoritma Isodata merupakan hal yang tepat apabila diterapkan pada perusahaan *EIT* (Yu, et al., 2012).

Pada penelitian yang dilakukan oleh Liu, dkk (2012), membahas tentang pemilihan fitur dari data ekspresi gen berdimensi tinggi. Dalam penelitian ini, data yang digunakan adalah data dalam bidang bioinformatika. Analisis data yang dilakukan adalah analisis data profil ekspresi gen leukemia akut dengan 50 penanda gen yang difilter berdasarkan indeks rasio *noise*. Dalam penelitian ini, metode yang diusulkan adalah algoritma *Fuzzy Isodata*. Algoritma Isodata menghitung diterapkan untuk menghitung tingkat kepekaan fitur. Evaluasi hasil dalam penelitian ini, dilakukan dengan 5 *microarray dataset* diantaranya, *DLBCL*, *Acute Leukemia*, *Multiple myeloma*, *Colon* dan *Prostate*. Hasil yang diperoleh menunjukkan bahwa fitur yang dipilih memiliki kinerja yang baik dalam melakukan klasifikasi dan klasterisasi pada 5 gen dataset profil ekspresi, sehingga dalam penelitian ini disimpulkan bahwa percobaan yang dilakukan dengan *Isodata* RFE mampu dalam memilih fitur dengan klasifikasi yang kuat dan kemampuan klasterisasi (Liu, et al., 2012).

Berdasarkan pemaparan beberapa penelitian sebelumnya, metode *K-Means* dan *Isodata* dapat diterapkan pada berbagai macam studi kasus penelitian. Dalam penelitian ini metode *K-Means* dan *Isodata* akan diterapkan pada studi kasus Klasterisasi data kejadian titik api wilayah Sumatera pada tahun 2001 hingga 2014. Tabel 2.1 menyajikan ringkasan kajian pustaka jurnal penelitian dengan metode *K-Means* dan *Isodata* yang telah dipaparkan sebelumnya. Tabel 2.1 menyajikan ringkasan Kajian pustaka yang dijabarkan berdasarkan judul, objek/parameter, metode, hasil dan nama jurnal.

Tabel 2.1 Kajian Pustaka

No	Judul	Objek/ Parameter	Metode	Hasil	Nama Jurnal
1.	Penerapan Metode <i>K-Means</i> untuk <i>Clustering</i> Produk <i>Online Shop</i> (Muningsih & Kiswati, 2015)	kode produk, jumlah transaksi, volume penjualan dan rata rata penjualan	K-Means	<i>cluster</i> 1 mencapai 80%, <i>cluster</i> 2 mencapai 60%, dan <i>cluster</i> 3 mencapai 80 %	Jurnal Bianglala Informatika

Tabel 2.1 Kajian Pustaka (lanjutan)

No	Judul	Objek/ Parameter	Metode	Hasil	Nama Jurnal
2.	<i>Comperative Analisis of K-Means and Fuzzy C-Means Algorithms</i> (Ghosh & Dubey, 2013)	<i>Dataset</i>	K-Means dan Fuzzy C-Means Algorithms	Algoritme <i>K-means</i> menghasilkan waktu 0.443755 detik sedangkan pada algoritme FCM menghasilkan waktu 0.781679 detik	IJACSA
3.	Implementasi Algoritme <i>K-Means</i> dalam Pengklasteran Mahasiswa Pelamar Beasiswa (Rohmawati, et al., 2015)	Data pelamar beasiswa BBM	<i>K-Means</i>	nilai purity hasil cluster algoritma <i>k-means</i> sebesar 0.806 atau 80.56%	JITTER
4.	Analisis Pengelompokan Jumlah Penumpang Bus Trans Jogja menggunakan Metode <i>Clustering K-Means</i> dan <i>Agglomerative Hierarchical Clustering (AHC)</i> (Zahrotun, 2015)	Data Penumpang Bus Trans Jogja	<i>K-Means</i> dan <i>AHC</i>	jumlah klaster 0 sebanyak 15 data, klaster 1 sebanyak 44 data, dan klaster 2 sebanyak 54 data.	Jurnal Informatika

Tabel 2.1 Kajian Pustaka (lanjutan)

No	Judul	Objek/ Parameter	Metode	Hasil	Nama Jurnal
5.	<i>An Improved Fuzzy Isodata Algorithm for Credit Risk Assessment of the EIT Enterprises</i> membahas tentang algoritma <i>Fuzzy Isodata</i> (Yu, et al., 2012)	4 indikator utama dan 15 indikator sekunder	<i>Fuzzy Isodata</i>	menghasilkan 10 investigasi perusahaan <i>EIT</i> harus ditetapkan menjadi 5 kredit nilai	Scientific Research
6.	<i>Feature selection based on sensitivity analysis of fuzzy isodata</i> (Liu, et al., 2012)	<i>DLBCL, Acute Leokimian, Multiple myeloma, Colon dan Prostate</i>	<i>fuzzy Isodata</i>	Hasil yang diperoleh menunjukkan bahwa fitur yang dipilih memiliki kinerja yang baik dalam melakukan klasifikasi dan klasterisasi pada 5 gen dataset profil ekspresi	Elsevier

2.1 The Earth Observation System Data and Information System (EOSDIS)

The Earth Observation System Data and Information System (EOSDIS) adalah pusat penampungan data ilmu bumi *National Aeronautics and Administration (NASA)*. *EOSDIS* memiliki kemampuan *end to end* untuk mengelola data ilmu bumi *NASA* dari berbagai sumber seperti satelit, pesawat, pengukuran di lapangan dan berbagai sumber lainnya. *EOSDIS* memiliki beberapa kemampuan diantaranya, memberikan komando dan melakukan kontrol, melakukan penjadwalan, *data capture* dan pengolahan awal data pada level 0. Kemampuan yang dimiliki *EOSDIS* merupakan bagian dari Misi

Operasi *EOSDIS* yang dikelola oleh *Earth Science Mission Operations (ESMO) project* (EOSDIS, n.d.).

Kemampuan lain yang dimiliki *EOSDIS* adalah *EOSDIS Science Operations, Earth Science Data and Information System (EOSDIS) project*. Kemampuan ini merupakan kemampuan pada generasi yang lebih tinggi pada level 1-4 data dari produk sains untuk misi *EOS*, *EOSDIS* dapat melakukan pengarsipan dan distribusi produk data dari *EOS* dan misi satelit lainnya, serta dari pesawat dan pengukuran di lapangan. Operasi ilmu *EOSDIS* dilakukan dalam sistem terdistribusi dari banyak *node* yang saling berhubungan atau yang lebih dikenal dengan "*Science Investigator-led Processing Systems (SIPS)*" dan mendistribusikan serta mendisiplinkan secara spesifik pada ilmu bumi yang lebih dikenal dengan "*Distributed Active Archive Center (DAACS)*". *DAACS* melayani komunitas pengguna yang besar dan beragam seperti yang ditunjukkan oleh *EOSDIS* metrik kinerja dengan menyediakan kemampuan untuk pencarian data dan akses ilmu data serta layanan khusus (EOSDIS, n.d.).

2.2 Min max normalisasi

Min Max normalisasi merupakan suatu konsep untuk mengubah skala atribut data menjadi lebih kecil. Data yang ternormalisasi dengan konsep min max normalisasi menghasilkan nilai dalam range 0 hingga 1. Tujuan dilakukannya normalisasi adalah untuk menyetarakan range seluruh variabel data yang diolah (Santoso, et al., 2017). Rumus min max normalisasi ditunjukkan pada persamaan 2.1.

$$v' = \frac{v - \min_A}{\max_A - \min_A} \quad (2.1)$$

Keterangan:

- v' : Nilai yang sudah ternormalisasi
- v : Nilai yang belum ternormalisasi
- \min_A : Nilai minimum data dari atribut ke A
- \max_A : Nilai maksimum data dari atribut ke A

2.3 Algoritme K-Means

K-Means merupakan salah satu algoritme klusterisasi yang dapat membagi data ke dalam beberapa kluster. Algoritme *K-Means* dapat mempartisi data pada kelompok sehingga data yang memiliki karakteristik yang sama dikumpulkan dalam satu kelompok atau yang lebih dikenal dengan kluster. Sebaliknya data yang memiliki karakteristik yang berbeda akan dimasukkan ke dalam kelompok yang lain. Dalam hal ini, tujuan dari pengelompokan data adalah untuk meminimalisasi fungsi objektif yang diatur dalam proses pengelompokan (Lloyd & Forgy, 1965).

Pada dasarnya algoritme *K-Means* melakukan dua proses yaitu pendeteksian lokasi pusat kluster dan proses pencarian anggota dari tiap-tiap kluster. Berikut merupakan proses dari algoritme *K-Means*:

1. Menentukan k sebagai jumlah kluster yang akan dibentuk dan menetapkan pusat kluster
2. Mengalokasikan data kedalam kluster secara acak
3. Menghitung pusat kluster dengan menggunakan persamaan

$$C_{kj} = \frac{\sum_{i=1}^p x_{ij}}{p} \quad (\text{Lloyd \& Forgey, 1965}) \quad (2.1)$$

Keterangan:

C_{kj} = Fitur ke j pada pusat kluster k

x_{ij} = Fitur ke j pada data ke i

p = Jumlah data

4. Menghitung jarak setiap data ke pusat kluster dengan persamaan *Euclidean distance*

$$d_{ik} = \sqrt{\sum_j^m (x_{ij} - C_{kj})^2} \quad (\text{Lloyd \& Forgey, 1965}) \quad (2.2)$$

Keterangan:

d_{ik} = Jarak data i ke pusat kluster k

m = Jumlah fitur

x_{ij} = Fitur ke j pada data ke i

C_{kj} = Fitur ke j pada pusat kluster k

5. Jika terdapat data yang berpindah kluster atau, atau jika perubahan nilai pada fungsi objektif masih diatas nilai ambang maka dilakukan iterasi dimulai dari langkah tiga.

2.4 Algoritme *Isodata*

Algoritme *Isodata* merupakan salah satu algoritme klasterisasi yang memungkinkan jumlah kluster disesuaikan secara otomatis berdasarkan proses iterasi yang dilakukan. Berbeda halnya dengan algoritme *K-Means*, Algoritme *Isodata* menentukan jumlah kluster pada tahap awal dan jumlah kluster tidak dapat berubah sepanjang iterasi yang dilakukan. Algoritme *Isodata* bekerja dengan menggabungkan kluster yang sama dan data pada suatu kluster dapat membelah berdasarkan simpangan baku yang besar (Ball & Hall, 1965). Berikut merupakan parameter yang digunakan dalam Algoritme *Isodata*.

K_0 = Kluster awal

n_{min} = Jumlah sampel minimum pada setiap kluster

σ^2_{max} = Varian maksimum (untuk pembelahan kluster)

d_{min} = Jarak berpasangan minimum (untuk penggabungan kluster)

Berikut merupakan tahapan algoritme *Isodata*

1. Mengalokasikan data pada kluster secara acak dan menghitung nilai rata-rata vektor $\{m_1, \dots, m_k\}$ dari data set.

- Menentukan titik data x untuk kluster dengan nilai rata-rata terdekat menggunakan Persamaan 2.3.

$$x \in \omega_i \text{ if } d(x, m_i) = \min\{d(x, m_1), \dots, d(x, m_k)\} \text{ (Ball \& Hall, 1965)} \quad (2.3)$$

Keterangan:

x = data

m = Pusat Kluster

- Menghilangkan kluster yang hanya memiliki sedikit anggota yaitu jika $n_j < n_{min}$ maka buang kluster dan memindahkan data pada kluster lain.
- Melakukan pembaruan nilai rata rata vektor pada kluster $\omega_j (j = 1, \dots, K)$ dengan Persamaan

$$m_j = \frac{1}{n_j} \sum_{x \in \omega_j} x, \text{ (Ball \& Hall, 1965)} \quad (2.4)$$

dan matrik kovarian pada persamaan 2.5

$$\Sigma_j = \frac{1}{n_j} \sum_{x \in \omega_j} (x - m_j)(x - m_j)^T \text{ (Ball \& Hall, 1965)} \quad (2.5)$$

Keterangan:

ω_j = kluster ke j

Elemen-elemen diagonal adalah varian $\sigma_1^2, \dots, \sigma_N^2$ sebanyak N dimensi

- Jika $K \leq K_0/2$, maka lakukan langkah 6 untuk melakukan pemisahan Kluster dan jika $K > 2K_0$ maka lakukan langkah 7 untuk melakukan penggabungan kluster.
- Membagi setiap kluster $\omega_j (j = 1, \dots, K)$, dan menghitung kovarian terbesar dengan persamaan 2.6.

$$\sigma_m^2 = \max\{\sigma_1^2, \dots, \sigma_N^2\}, \text{ Jika } \sigma_m^2 > \sigma_{max}^2 \text{ dan } n_j > 2n_{min} \text{ (Ball \& Hall, 1965)} \quad (2.6)$$

maka membagi dua pusat kluster baru,

$$m_j^+ = m_j + \sigma_m, \quad m_j^- = m_j - \sigma_m,$$

Atau menggunakan PCA untuk mencari varian sesuai dengan *Eigen value* terbesar λ_{max} dan membagi kluster berdasarkan vektor *Eigen*.

- Menggabungkan kluster $K(K-1)/2$ dengan menggunakan *Bhattacharyya Distance* pada persamaan 2.7.

$$d_B(\omega_i, \omega_j) = \frac{1}{4}(m_i - m_j)^T \left[\frac{\Sigma_i + \Sigma_j}{2} \right]^{-1} (m_i - m_j) + \log \left[\frac{\frac{\Sigma_i + \Sigma_j}{2}}{(|\Sigma_i| |\Sigma_j|)^{1/2}} \right],$$

$$(1 \leq i, j \leq K, i > j) \text{ (Ball \& Hall, 1965)} \quad (2.7)$$

Penggabungan klaster, dapat menggunakan metode yang berbeda. Apa bila hasil d_B memenuhi kondisi $d_B(\omega_i, \omega_j) < d_{min}$, maka kedua klaster tersebut, digabungkan dengan Persamaan 2.8

$$m_i = \frac{1}{n_1 + n_j} [n_1 m_i + n_j m_j] \quad (\text{Ball \& Hall, 1965}) \quad (2.8)$$

hapus m_j , set $K \leftarrow K - 1$

8. Kembali pada langkah 2 apabila jumlah iterasi maksimum belum tercapai, Jika iterasi maksimum tercapai, maka berhenti melakukan iterasi.

2.5 Silhouette Coefficient

Silhouette Coefficient (SC) merupakan sebuah metode untuk menghitung nilai objek yang berada pada suatu klaster. Metode ini merupakan metode gabungan dari *cohesion* dan *separation* (Wahyuni, et al., 2016). Berikut merupakan proses yang dilakukan dalam metode *Silhouette Coefficient*.

1. Menghitung rata-rata jarak antar objek yang berada dalam satu klaster yang sama, dengan persamaan 2.9.

$$a(i) = \frac{1}{|A|} \sum_{j \in C} d(i, j) \quad (2.9)$$

Keterangan:

j = objek lain dalam suatu klaster yang sama

$d(i, j)$ = jarak antar objek i dengan j

2. Menghitung jarak antar objek yang berbeda klaster dan mengambil nilai yang terkecil.

$$d(i, C) = \frac{1}{|A|} \sum_{j \in C} d(i, j) \quad (2.9)$$

Keterangan:

$d(i, C)$ = jarak rata-rata antar objek i dengan seluruh objek C dimana

$A \neq C$

$b(i) = \min_{C \neq A} d(i, C)$

3. Menghitung nilai *Silhouette Coefficient* dengan menggunakan persamaan

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (\text{Rousseeuw, 1987}) \quad (3.0)$$

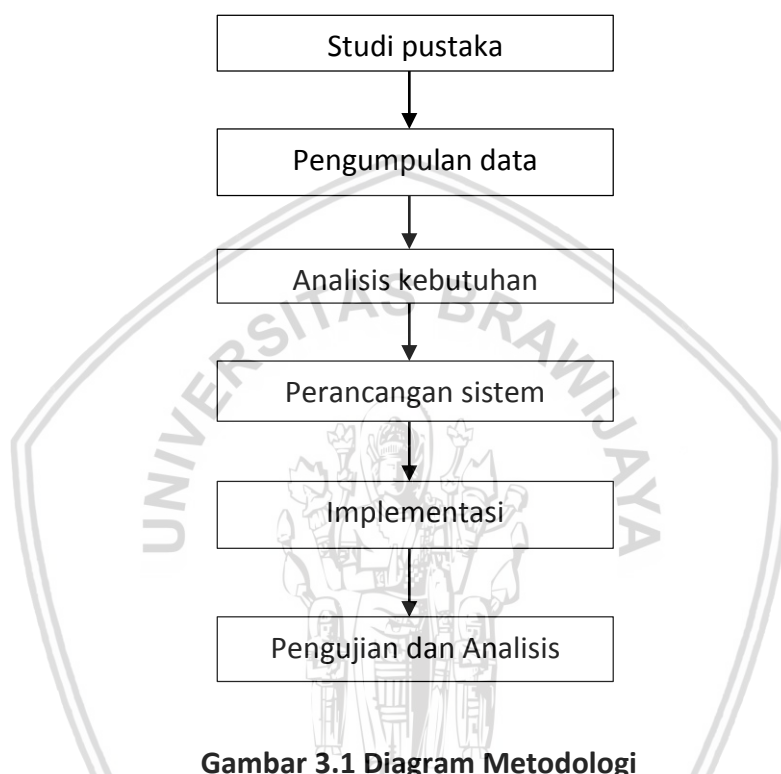
Keterangan:

$s(i)$ = hasil akhir penentuan nilai *Silhouette Coefficient*.

Nilai yang diperoleh dari metode pengujian SC dalam kisaran -1 hingga 1. Semakin nilai SC mendekati angka 1 maka semakin baik kualitas klaster yang dihasilkan, akan tetapi jika nilai SC mendekati angka -1 maka kualitas klaster yang dihasilkan semakin buruk.

BAB 3 METODOLOGI DAN PERANCANGAN

Pada bab metodologi akan dibahas tentang tahapan sistematika dalam penyelesaian penelitian. Dalam subbab metodologi akan dibahas tentang studi pustaka, pengumpulan data, analisis kebutuhan, perancangan sistem, Implementasi, Pengujian dan Analisis. Gambar 3.1 merupakan diagram alir metodologi penelitian yang digunakan.



Gambar 3.1 Diagram Metodologi

3.1 Studi pustaka

Studi pustaka merupakan tahapan untuk menemukan sumber kajian pustaka yang nantinya akan dijadikan acuan dalam penelitian ini. Kajian pustaka yang digunakan dalam penelitian ini didapat dari berbagai macam sumber seperti jurnal ilmiah, *website*, buku dan media Elektronik. Landasan kepustakaan dalam penelitian ini dapat berupa penelitian terdahulu, dasar teori, objek penelitian yang digunakan.

3.2 Pengumpulan data

Data yang digunakan adalah data kejadian titik api pada wilayah Sumatera. Data didapat dari situs EARTHDATA dengan alamat web earthdata.nasa.gov. Data Kejadian titik api merupakan data yang bersifat publik dan dapat diperoleh secara gratis. Gambar 3.2 merupakan peta wilayah yang diseleksi untuk studi kasus data kejadian titik api pada wilayah Sumatera.



Gambar 3.2 Peta Seleksi Wilayah

Gambar 3.2 memvisualisasikan peta wilayah yang di seleksi, yaitu pulau Sumatera. Pada Gambar 3.3 merupakan data yang didapat dari hasil *crop* wilayah pada Gambar 3.2.

0101000000E7FBA9F1D29559409A999999999D13F																			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	geom	latitude	longitude	brightness	scan	track	acq_date	acq_time	satellite	confidence	version	bright_t31	frp						
2	0101000000105839	0.58	101.981	311.3	3	1.6	12/10/2005	308	Terra	35	5.1	286.9	42.3						
3	01010000000A0D7A	0.568	102.16	312.5	1	1	12/13/2005	339	Terra	51	5.1	293.1	9.9						
4	0101000000E7FBA9	0.275	102.341	315.7	1	1	12/13/2005	339	Terra	52	5.1	292.6	10.4						
5	0101000000048E17A	-2.517	104.245	323.6	1.2	1.1	12/13/2005	340	Terra	77	5.1	286.2	27						
6	0101000000034D6E	-3.693	104.396	317	1.3	1.1	12/13/2005	340	Terra	40	5.1	291.8	16.6						
7	0101000000000000	-3.721	104.375	350.3	1.3	1.1	12/13/2005	340	Terra	95	5.1	289.5	92.9						
8	01010000000621058	-3.722	104.386	321.2	1.3	1.1	12/13/2005	340	Terra	65	5.1	286.4	24.8						
9	010100000003F355E	-4.546	104.317	312.6	1	1	12/13/2005	632	Aqua	55	5.1	285.3	10						
10	01010000000CF753E	-4.547	104.307	312.7	1	1	12/13/2005	632	Aqua	55	5.1	285.4	10.4						
11	01010000000B81E85	-3.719	104.38	318.9	1	1	12/13/2005	632	Aqua	59	5.1	288.6	14						
12	0101000000052B81E	-3.594	104.28	311	1	1	12/13/2005	632	Aqua	31	5.1	285	7.4						
13	01010000000A01A2	-3.481	103.471	315.4	1.1	1	12/13/2005	632	Aqua	60	5.1	292.8	10.1						
14	0101000000084C87E	-2.817	103.541	315.6	1.1	1	12/13/2005	632	Aqua	44	5.1	292.7	10.5						
15	01010000000894160	-1.396	102.544	310.2	1.2	1.1	12/13/2005	633	Aqua	24	5.1	274.8	12.4						
16	010100000005EBA4F	0.553	102.297	317	1.1	1.1	12/13/2005	633	Aqua	41	5.1	295.5	15						
17	010100000000B9F71	-0.536	102.574	332.1	3.1	1.7	12/14/2005	1505	Terra	100	5.1	271	226.6						
18	01010000000A01A2	-0.533	102.596	308.2	3.1	1.7	12/14/2005	1505	Terra	58	5.1	274.2	73.1						
19	0101000000023DBF1	-0.537	102.569	322.3	3.1	1.7	12/14/2005	1505	Terra	100	5.1	269.6	153.3						
20	0101000000008AC1C	-2.902	105.053	322.8	1.3	1.1	12/15/2005	620	Aqua	76	5.1	287.6	31.8						
21	01010000000068195	1.52	101.196	312.3	2.2	1.4	12/16/2005	409	Terra	47	5.1	292.1	31.2						
22	01010000000E3A59E	1.591	99.877	314.8	1.8	1.3	12/16/2005	409	Terra	62	5.1	290.5	22.8						
23	01010000000FED47E	1.589	99.893	312.2	1.8	1.3	12/16/2005	409	Terra	48	5.1	289.4	17.3						

Gambar 3.3 Data Kejadian Titik Api

Gambar 3.3 merupakan data kejadian titik api berdasarkan hasil seleksi peta yang tampak pada Gambar 3.2. Data pada Gambar 3.2 merupakan data dalam format CSV. Data kejadian titik api yang pilih terhitung dari tanggal 1 Januari 2001 hingga 31 Desember 2014. Data kejadian titik api yang diperoleh terdiri dari 289,257 baris data dan 13 kolom. Variable 13 kolom yang didapat adalah *geom*, *latitude*, *longitude*, *brigness*, *scan*, *track*, *acq_date*, *acq_time*, *satellite*, *confidence*, *version*, *bright_t31*, *frp*. Berdasarkan 13 variable yang didapat dipilih lima variable yang akan di klaster yaitu *latitude*, *longitude*, *brightness*, *acq_date*, *convidence*.

Latitude dalam bahasa Indonesia adalah garis bujur. Garis bujur adalah garis khayal yang ditarik dari kutub utara ke kutub selatan. Garis bujur dapat membagi bumi menjadi dua bagian, diantaranya bumi bagian timur dan bumi bagian barat.

Garis bujur dapat menentukan sebuah lokasi tertentu. Hal ini, dikombinasikan dengan adanya garis lintang. Dalam koordinat garis lintang digambarkan dengan sumbu x, sedangkan garis bujur dilambangkan sebagai sumbu y. Garis bujur juga dapat dijadikan sebagai dasar dalam menentukan pembagian waktu di seluruh bagian bumi (Sari, 2016).

Longitude dalam bahasa Indonesia adalah garis lintang. Garis lintang merupakan garis yang biasa digunakan dalam menentukan suatu lokasi yang berpusat pada garis khatulistiwa. Garis lintang mengitari bumi dari bagian ekuator hingga pada bagian kutub utara dan kutub selatan. Garis yang berada di sebelah selatan ekuator dikenal dengan istilah Lintang Selatan (LS), sedangkan garis yang berada di sebelah utara ekuator dikenal dengan istilah Lintang Utara (LS). Jarak antara garis satu dengan yang lainnya dihitung dengan satuan derajat. Dalam ilmu geografi, satuan derajat juga disebut sebagai jam. 1 yang terbagi menjadi 60 menit dan 1 menit terbagi menjadi 60 detik. Sehingga dalam hal ini, garis lintang pada suatu lokasi dapat juga disebut sebagai jam (Sari, 2016).

Brightnees dalam bahasa Indonesia adalah kecerahan atau dengan makna lain adalah tingkat kecerahan titik api yang ada pada suatu wilayah. Variabel *brightness* digunakan sebagai indikator untuk menentukan seberapa parah kejadian titik api pada suatu wilayah.

Acq_date merupakan variabel dalam data kejadian titik api yang menjabarkan data tanggal, bulan, dan tahun kejadian titik api dalam studi kasus ini. *Acq_date* dapat digunakan untuk menemukan pola waktu kejadian titik api, sehingga variabel *acq_date* diasumsikan sebagai variabel yang penting untuk di klaster. Tipe data parameter *Acq_date* terformat dalam bentuk *Date*. Data pada parameter *Acq_date* dikonversi dalam bentuk number berdasarkan nomer seri yang dikenali Excel sebagai tanggal. Sebagai contoh rumus =DATEVALUE ("1/1/2008") mengembalikan 39448, sebagai nomer seri tanggal 1/1/2008. Hasil konversi format tanggal pada nomer seri memungkinkan hasil yang berbeda pada masing-masing komputer. Hal ini bergantung dengan pengaturan tanggal pada masing-masing komputer (Microsof, n.d.).

Convidence dalam bahasa Indonesia adalah keyakinan, dalam istilah adalah tingkat keyakinan untuk mengukur suatu kenyataan atau kejadian. Dalam studi kasus ini *convidence* digunakan sebagai indikator untuk mengukur tingkat keyakinan kejadian titik api di beberapa wilayah.

Sebelum proses pengklasteran, data dari *latitude*, *Longitude*, *Brightnees*, *Acq_date* dinormalisasi dengan menggunakan konsep min max normalisasi. Sehingga, data angka yang diolah merupakan data dalam *range* 0 hingga 1.

3.3 Analisis kebutuhan

Analisis kebutuhan merupakan tahapan untuk mendefinisikan kebutuhan kebutuhan yang terkait untuk menyelesaikan penelitian yang bersangkutan. Kebutuhan yang didefinisikan dapat berupa kebutuhan perangkat keras,

perangkat lunak maupun kebutuhan data. Berikut merupakan rincian kebutuhan yang terkait dalam menyelesaikan penelitian ini.

3.3.1 Perangkat keras

Pada tahap pengerjaan penelitian *K-Means* dan *Isodata* untuk Klasterisasi data kejadian titik api wilayah Sumatera pada tahun 2001 hingga 2014 membutuhkan perangkat keras yang mendukung. Tabel 3.1 berikut merupakan rincian spesifikasi perangkat keras pendukung implementasi sistem.

Tabel 3.1 Rincian kebutuhan Perangkat Keras

No	Perangkat	Spesifikasi
1	Processor	Intel(R) Core(TM) i3-2350 CPU @2.30GHz 2.30GHz
2	Memory RAM	4.00 GB
3	Hard Disk	465.76 GB
4	VGA	Intel(R) HD Graphics Family

3.3.2 Perangkat lunak

Adapun Perangkat lunak yang digunakan dalam pengerjaan penelitian ini dijabarkan pada tabel 3.2.

Tabel 3.2 Rincian pendukung perangkat lunak

No	Perangkat	Spesifikasi Perangkat
1.	Sistem Operasi	Windows 8.1 Pro 64-bit
2.	Tools Dokumentasi	Microsoft Office 2013
3.	Tools Pemrograman	NetBeans IDE 8.0.2
4.	Paket Pemrograman Java	Java SE Development Kit update 131

3.4 Perancangan sistem

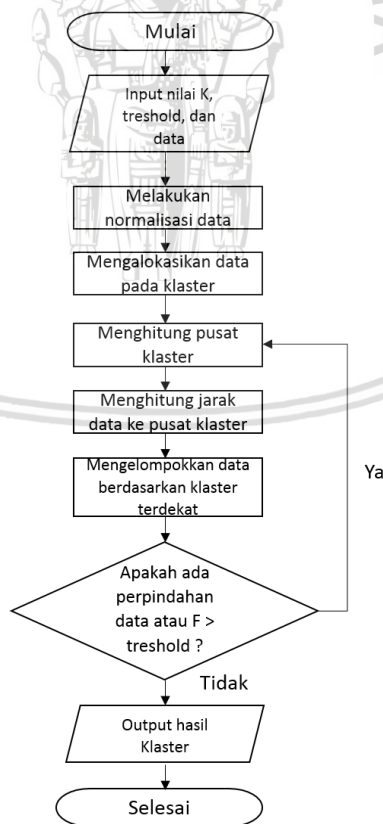
Perancangan sistem merupakan tahapan untuk melakukan pemodelan berdasarkan studi kasus yang diangkat dalam penelitian ini. Pemodelan dilakukan dengan pembuatan diagram alir, manualisasi algorithme *K-Means* dan manualisasi algorithme *Isodata*.

3.4.1 Algorithme *K-Means*

Algorithme *K-Means* merupakan salah satu algorithme yang akan terapkan dalam penelitian ini. Perancangan sistem pada subbab 3.41 ini menggambarkan rancangan dengan menggunakan diagram alir untuk proses algorithme *K-Means*.

Pada algoritme *K-Means* terdapat beberapa tahapan yang akan diproses. Pada tahapan yang pertama ketika sistem dijalankan yang maka langkah yang dilakukan pengguna adalah menentukan nilai k , *threshold* dan masukan data titik panas atau data kejadian titik api yang akan di oleh dengan algoritme *K-Means*.

Pada tahapan yang kedua sistem melakukan normalisasi data kejadian titik api yang telah dimasukkan. Metode yang digunakan dalam melakukan normalisasi adalah *min max* normalisasi. Setelah data dinormalisasi, tahap yang ketiga adalah sistem mengalokasi data pada masing-masing kluster secara acak. Pada tahapan yang keempat, sistem menghitung pusat kluster atau yang lebih dikenal dengan istilah *Centroid* pada masing-masing kluster yang telah dialokasikan sebelumnya. Pada tahapan kelima, sistem menghitung jarak data ke seluruh pusat kluster menggunakan persamaan *Euclidean distance*. Setelah nilai jarak data ke seluruh pusat kluster didapat, maka masuk pada tahapan keenam yaitu mengelompokkan data berdasarkan jarak data ke pusat kluster terdekat. Pada tahapan ini data dapat berpindah kluster berdasarkan jarak terpendek data dengan pusat kluster terdekat. Pada tahapan yang ketujuh sistem menyeleksi dengan kondisi, jika terdapat perpindahan data ke kluster lain atau nilai $F > \text{threshold}$ maka sistem akan kembali pada tahapan yang keempat. Pada kondisi lainnya proses sistem akan berhenti sehingga hasil pengelompokan akhir sistem dengan Algoritme *K-Means* dapat ditetapkan. Gambar 3.4 merupakan diagram alir pemodelan algoritme *K-Means* pada studi kasus klusterisasi kejadian titik api.



Gambar 3.4 Diagram alir Algoritme *K-Means*

Diagram alir pada Gambar 3.4 merupakan tahapan penyelesaian dari algoritme *K-Means* dengan tahapan, (1) menentukan jumlah kluster, (2) menghitung jarak data ke pusat kluster, (3) mengelompokkan data ke dalam kluster, (4) menghitung pusat kluster baru dan melakukan pengecekan data. Jika terdapat data yang berpindah kluster, maka akan dilakukan iterasi yang dimulai dari langkah kedua dan apabila tidak, proses akan selesai.

3.4.2 Manualisasi alortime *K-Means*

Manualisasi alortime *K-Means* merupakan tahapan perhitungan manual dari penyelesaian alortime *K-Means* pada studi kasus klusterisasi data kejadian titik api. Dalam tahapan manualisasi, data sampel yang digunakan adalah data sampel kejadian titik api sebanyak sepuluh data. Tabel 3.3 merupakan data sampel yang digunakan dalam proses manualisasi algoritme *K-Means*.

Tabel 3.3 Data sampel kejadian titik api

No	Latitude	Longitude	Brigness	Acq_Date	Convidence
1	0.58	101.981	311.3	12/10/2005	35
2	0.568	102.16	312.5	12/11/2005	51
3	0.275	102.341	315.7	12/12/2005	52
4	-2.517	104.245	323.6	12/13/2005	77
5	-3.693	104.396	317	12/14/2005	40
6	-3.721	104.375	350.3	12/15/2005	95
7	-3.722	104.386	321.2	12/16/2005	65
8	-4.546	104.317	312.6	12/17/2005	55
9	-4.547	104.307	312.7	12/18/2005	55
10	-3.719	104.38	318.9	12/19/2005	53

Sebelum dilakukan proses perhitungan manual, data pada Table 3.3 dilakukan normalisasi. Berikut merupakan normalisasi data sampel dengan menggunakan persamaan 3.1.

$$\begin{aligned}
 \text{Normalisasi} &= \frac{(X - \text{Min})}{(\text{Max} - \text{Min})} \\
 &= \frac{0.58 - (-4.547)}{0.58 + 4.547} \\
 &= 1
 \end{aligned} \tag{3.1}$$

Data yang dihitung pada persamaan 3.1 merupakan data pada baris pertama kolom pertama sebagai sampel untuk menghitung nilai normalisasi dengan menggunakan *min max* normalisasi. Nilai data yang digunakan adalah $x = 0.58$ yang menghasilkan nilai normalisasi = 1. Pada Tabel 3.4 menjabarkan seluruh hasil normalisasi data sampel kejadian titik api yang digunakan.

Tabel 3.4 Normalisasi data sampel *K-Means*

No	F1	F2	F3	F4	F5
1	1	0	0	0	0
2	0.99765945	0.074120083	0.03076923	0.11111111	0.266666667
3	0.94051102	0.149068323	0.11282051	0.22222222	0.283333333
4	0.395943047	0.93747412	0.31538462	0.33333333	0.7
5	0.166569144	1	0.14615385	0.44444444	0.083333333
6	0.16110786	0.991304348	1	0.55555556	1
7	0.160912815	0.995859213	0.25384615	0.66666667	0.5
8	0.000195046	0.967287785	0.03333333	0.77777778	0.333333333
9	0	0.963146998	0.03589744	0.88888889	0.333333333
10	0.161497952	0.993374741	0.19487179	1	0.3

Pada Table 3.4 fitur *latitude* diinisialisasi sebagai F1, fitur *longitude* sebagai f2, fitur *brightness* sebagai f3, fitur *acq_date* sebagai F4 dan fitur *convidence* sebagai F5. Hasil normalisasi data pada Tabel 3.3 berada pada interval antara 0-1. Data hasil normalisasi pada Tabel 3.4 yang akan digunakan dalam proses perhitungan manual algoritme *K-Means*. Berikut merupakan tahapan perhitungan manual algoritme *K-Means*.

1. Menentukan K sebagai jumlah kluster yang akan dibentuk $K=3$
2. Mengalokasikan data pada tiap K

Tabel 3.5 Alokasi data pada K -Means

No	F1	F2	F3	F4	F5	K 1	K 2	K 3
1	1	0	0	0	0	*		
2	0.99765945	0.074120083	0.03076923	0.11111111	0.266666667		*	
3	0.94051102	0.149068323	0.11282051	0.22222222	0.283333333		*	
4	0.395943047	0.93747412	0.31538462	0.33333333	0.7	*		
5	0.166569144	1	0.14615385	0.44444444	0.083333333		*	
6	0.16110786	0.991304348	1	0.55555556	1			*
7	0.160912815	0.995859213	0.25384615	0.66666667	0.5	*		
8	0.000195046	0.967287785	0.03333333	0.77777778	0.333333333			*
9	0	0.963146998	0.03589744	0.88888889	0.333333333		*	

Tabel 3.5 Alokasi data pada KK -Means (lanjutan)

No	F1	F2	F3	F4	F5	K1	K2	K3
10	0.161497952	0.993374741	0.19487179	1	0.3		*	
Jumlah						3	5	2

Pada Tabel 3.5 menampilkan K sebagai kluster. Tabel 3.6 merupakan alokasi data pada kluster 1 pada masing-masing fitur.

Tabel 3.6 Alokasi data kluster 1 (K -Means)

No	K1F1	K1F2	K1F3	K1F4	K1F5
1	1	0	0	0	0
2					
3					
4	0.39594305	0.93747412	0.31538462	0.33333333	0.7
5					
6					
7	0.16091281	0.99585921	0.25384615	0.66666667	0.5
8					
9					
10					
Jumlah	1.55685586	1.93333333	0.56923077	1	1.2

Pada Tabel 3.6 menampilkan pengalokasian data pada k_1 berdasarkan data fitur yang dialokasikan. Tabel 3.7 merupakan alokasi data kluster 2 pada masing-masing fitur.

Tabel 3.7 Alokasi data kluster 2 (K -Means)

No	K2F1	K2F2	K2F3	K2F4	K2F5
1					
2	0.99765945	0.0741201	0.0307692	0.111111	0.266667
3	0.94051102	0.1490683	0.1128205	0.222222	0.283333
4					
5	0.166569144	1	0.1461538	0.444444	0.083333
6					
7					
8					
9	0	0.963147	0.0358974	0.888889	0.333333
10	0.161497952	0.9933747	0.1948718	1	0.3
Jumlah	2.266237566	3.1797101	0.5205128	2.666667	1.266667

Pada Tabel 3.7 menampilkan pengalokasian data pada k2 berdasarkan data fitur yang dialokasikan. Tabel 3.8 merupakan alokasi data kluster 3 pada masing-masing fitur.

Tabel 3.8 Alokasi data kluster 3 (*K-Means*)

No	K3F1	K3F2	K3F3	K3F4	K3F5
1					
2					
3					
4					
5					
6	0.161108	0.991304	1	0.555556	1
7					
8	0.000195	0.967288	0.033333	0.777778	0.333333
9					
10					
Jumlah	0.161303	1.958592	1.033333	1.333333	1.333333

Pada Tabel 3.8 menampilkan pengalokasian data pada k3 berdasarkan data fitur yang dialokasikan.

- Menghitung pusat kluster data yang ada pada masing-masing kluster menggunakan persamaan 2.1.

$$C_{kj} = \frac{1.55685586}{3}$$

$$= 0.518951954$$

Sehingga diperoleh hasil yang ditampilkan pada Tabel 3.9:

Tabel 3.9 Pusat kluster (*K-Means*)

Kel	Centroid F1	Centroid F2	Centroid F3	Centroid F4	Centroid F5
1	0.518951954	0.644444444	0.18974359	0.333333333	0.4
2	0.453247513	0.635942029	0.10410256	0.533333333	0.253333333
3	0.080651453	0.979296066	0.51666667	0.666666667	0.666666667

Pada Tabel 3.9 menampilkan *centroid* sebagai pusat kluster yang dijabarkan berdasarkan fiturnya.

- Menghitung Jarak data ke *centroid* dengan persamaan *Euclidean Distance*

$$d_{ik} = \sqrt{(1 - 0.518951954)^2 + (0 - 0.644444444)^2}$$

$$+ (0 - 0.18974359)^2 + \sqrt{(0 - 0.333333333)^2 + (0 - 0.4)^2}$$

$$= 0.976642005$$

Sehingga diperoleh hasil yang ditampilkan pada Tabel 3.10:

Tabel 3.10 Jarak data ke *Centroid* (K-Means)

No	Jarak Ke C1	Jarak Ke C2	Jarak Ke C3	Min	Klaster Baru	Klaster Sebelumnya
1	0.976642005	1.030931672	1.72048129	0.976642	1	1
2	0.804278612	0.89210729	1.537850942	0.80427861	1	2
3	0.674524646	0.756464049	1.391475334	0.67452465	1	2
4	0.454734262	0.615106999	0.5038795	0.45473426	1	1
5	0.604249055	0.503281471	0.731266536	0.50328147	2	2
6	1.14641282	1.253899959	0.603063291	0.60306329	3	3
7	0.613925578	0.562179082	0.321820794	0.32182079	3	1
8	0.774452083	0.621456288	0.603063291	0.60306329	3	3
9	0.841211927	0.67079661	0.63118803	0.63118803	3	2
10	0.839044968	0.664131432	0.59652482	0.59652482	3	2
Jumlah	2.910179525	0.503281471	2.755660227			

Pada Tabel 3.10 menampilkan tabel jarak data ke *centroid*. Baris pada kolom “Min” dan angka yang diarsir merupakan data jarak minimal diantara jarak data C1, C2 dan C3. Pembaruan klaster dapat dilakukan berdasarkan data “Min” yang telah diperbarui.

$$F \text{ Baru} = \text{Jumlah C1} + \text{Jumlah C2} + \text{Jumlah C3}$$

$$F \text{ Baru} = 2.910179525 + 0.503281471 + 2.755660227$$

$$= 6.169121223$$

$$\text{Delta} = 6.169121223 - 0$$

$$= 6.169121223$$

5. Pada iterasi Ke 0 terdapat perpindahan data dan nilai *delta* masih diatas nilai ambang yaitu $6.169121223 > 0.8$, maka pada tahapan berikutnya melakukan iterasi yang dimulai pada langkah 2

Iterasi ke 1

1. Mengalokasikan setiap data pada *centroid* terdekat seperti yang ditampilkan pada Tabel 3.11

Tabel 3.11 Alokasi data pada tiap K iterasi 1 (K-Means)

No	F1	F2	F3	F4	F5	K1	K2	K3
1	1	0	0	0	0	*		
2	0.99765945	0.074120083	0.03076923	0.111111111	0.266666667	*		
3	0.94051102	0.149068323	0.11282051	0.222222222	0.283333333	*		
4	0.395943047	0.93747412	0.31538462	0.333333333	0.7	*		
5	0.166569144	1	0.14615385	0.444444444	0.083333333		*	
6	0.16110786	0.991304348	1	0.555555556	1			*
7	0.160912815	0.995859213	0.25384615	0.666666667	0.5			*
8	0.000195046	0.967287785	0.033333333	0.777777778	0.333333333			*
9	0	0.963146998	0.03589744	0.888888889	0.333333333			*
10	0.161497952	0.993374741	0.19487179	1	0.3			*
Jumlah						4	1	5

Pada Tabel 3.11 menampilkan alokasi data kluster pada iterasi 1 berdasarkan hasil dari proses iterasi ke 0. Tabel 3.12 menyajikan alokasi data kluster 1 pada masing-masing fitur.

Tabel 3.12 Alokasi data kluster 1 Iterasi 1 (K-Means)

No	K1F1	K1F2	K1F3	K1F4	K1F5
1	1	0	0	0	0
2	0.99765945	0.07412008	0.030769231	0.11111111	0.2666667
3	0.94051102	0.14906832	0.112820513	0.22222222	0.2833333
4					
5					
6					
7					
8					
9					
10					
Jumlah	2.93817047	0.22318841	0.143589744	0.33333333	0.55

Pada Tabel 3.12 menampilkan alokasi data kluster 1 berdasarkan data fitur yang di alokasikan pada iterasi ke 1. Tabel 3.13 berikut menyajikan alokasi data kluster 2 pada masing-masing fitur.

Tabel 3.13 Alokasi data kluster 2 iterasi 1 (*K-Means*)

No	K2F1	K2F2	K2F3	K2F4	K2F5
1					
2					
3					
4	0.395943047	0.9374741	0.3153846	0.333333	0.7
5					
6					
7					
8					
9					
10					
Jumlah	0.395943047	0.9374741	0.3153846	0.333333	0.7

Pada Tabel 3.13 menampilkan alokasi data kluster 2 berdasarkan data fitur yang di alokasikan pada iterasi ke 1. Tabel 3.14 berikut menyajikan alokasi data kluster 3 pada masing-masing fitur.

Tabel 3.14 Alokasi data kluster 3 iterasi 1 (*K-Means*)

K3F1	K3F2	K3F3	K3F4	K3F5
0.166569	1	0.146154	0.444444	0.083333
0.161108	0.991304	1	0.555556	1
0.160913	0.995859	0.253846	0.666667	0.5
0.000195	0.967288	0.033333	0.777778	0.333333
0	0.963147	0.035897	0.888889	0.333333
0.161498	0.993375	0.194872	1	0.3
0.650283	5.910973	1.664103	4.333333	2.55

Pada Tabel 3.14 menampilkan alokasi data kluster 3 berdasarkan data fitur yang dialokasikan pada iterasi ke 1.

2. Menghitung pusat kluster dari data yang ada pada setiap kluster menggunakan persamaan 2.1

$$C_{kj} = \frac{2.93817047}{4}$$

$$= 0.734542618$$

Sehingga diperoleh hasil yang ditampilkan pada Tabel 3.15 berikut:

Tabel 3.15 Pusat kluster iterasi 1 (K-Means)

Kel	Centroid F1	Centroid F2	Centroid F3	Centroid F4	Centroid F5
1	0.734542618	0.055797101	0.03589744	0.083333333	0.1375
2	0.395943047	0.93747412	0.31538462	0.333333333	0.7
3	0.130056563	1.182194617	0.33282051	0.866666667	0.51

3. Menghitung Jarak data ke *centroid* dengan persamaan *Euclidian Distance*

$$\begin{aligned}
 d_{ik} &= \sqrt{(1 - 0.518951954)^2 + (0 - 644444444)^2 +} \\
 &\quad (0 - 0.18974359)^2 + \sqrt{(0 - 0.333333333)^2 + (0 - 0.4)^2} \\
 &= 0.976642005
 \end{aligned}$$

Sehingga diperoleh hasil yang ditampilkan pada Tabel 3.16 berikut:

Tabel 3.16 Jarak data ke *Centroid* iterasi 1 K-Means

No	Jarak Ke C1	Jarak Ke C2	Jarak Ke C3	Min	Kluster Baru	Kluster Sebelumnya
1	0.31736455	1.39438915	1.810073562	0.31736455	1	1
2	0.2950392	1.19398879	1.643737883	0.2950392	1	1
3	0.31240542	1.07010606	1.49641999	0.31240542	1	1
4	1.16147465	0	0.671887043	0	2	1
5	1.16602089	0.69121949	0.655506324	0.65550632	3	2
6	1.76079387	0.81616555	0.905219238	0.81616555	2	3
7	1.31603708	0.46211051	0.286373777	0.28637378	3	3
8	1.37502436	0.75434314	0.438011275	0.43801127	3	3
9	1.43181952	0.82378972	0.429842786	0.42984279	3	3
10	1.44892681	0.82283484	0.34285439	0.34285439	3	3
Jumlah	0.92480918	0.81616555	2.152588552			

Pada Tabel 3.16 menampilkan jarak data pada pusat kluster beserta penentuan jarak minimal data diantara ketiga kluster, sehingga dapat ditentukan pembaruan kluster baru dan kluster lama. Berikut merupakan perhitungan fungsi objektif.

F Baru = Jumlah C1+Jumlah C2 Jumlah C3

F Baru = 0.92480918 + 0.81616555 + 2.152588552
= 3.893563275

Delta = 6.169121223 – 3.893563275
= 3.893563275

6. Pada iterasi Ke 0 terdapat perpindahan data dan nilai delta masih diatas nilai ambang yaitu $3.893563275 > 0.8$, maka pada tahapan berikutnya melakukan iterasi yang dimulai pada langkah 2.

Iterasi 2

1. Mengalokasikan setiap data pada *centroid* terdekat seperti yang ditampilkan pada Tabel 3.17.

Tabel 3.17 Alokasi data pada tiap K iterasi 2 K-Means

No	F1	F2	F3	F4	F5	K1	K2	K3
1	1	0	0	0	0	*		
2	0.99765945	0.074120083	0.03076923	0.111111111	0.266666667	*		
3	0.94051102	0.149068323	0.11282051	0.222222222	0.283333333	*		
4	0.395943047	0.93747412	0.31538462	0.333333333	0.7		*	
5	0.166569144	1	0.14615385	0.444444444	0.083333333			*
6	0.16110786	0.991304348	1	0.555555556	1		*	
7	0.160912815	0.995859213	0.25384615	0.666666667	0.5			*
8	0.000195046	0.967287785	0.033333333	0.777777778	0.333333333			*
9	0	0.963146998	0.03589744	0.888888889	0.333333333			*
10	0.161497952	0.993374741	0.19487179	1	0.3			*
Jumlah						3	2	5

Pada Tabel 3.17 menampilkan alokasi data masing-masing kluster berdasarkan hasil dari proses iterasi ke 1. Tabel 3.18 merupakan alokasi data kluster 1 pada masing-masing fitur.

Tabel 3.18 Alokasi data kluster 1 iterasi 2 K-Means

No	K1F1	K1F2	K1F3	K1F4	K1F5
1	1	0	0	0	0
2	0.99765945	0.07412008	0.030769231	0.11111111	0.26666667
3	0.94051102	0.14906832	0.112820513	0.22222222	0.28333333
4					
5					
6					
7					
8					
9					
10					
Jumlah	2.93817047	0.22318841	0.143589744	0.33333333	0.55

Pada Tabel 3.18 menampilkan alokasi data kluster 1 berdasarkan fitur yang di alokasikan pada iterasi ke 2. Tabel 3.19 berikut merupakan alokasi data kluster 2 pada masing-masing fitur.

Tabel 3.19 Alokasi data kluster 2 iterasi 2 K-Means

No	K2F1	K2F2	K2F3	K2F4	K2F5
1					
2					
3					
4	0.39594305	0.9374741	0.3153846	0.333333	0.7
5					
6	0.16110786	0.9913043	1	0.555556	1
7					
8					
9					
10					
Jumlah	0.55705091	1.9287785	1.3153846	0.888889	1.7

Pada Tabel 3.19 menampilkan alokasi data kluster 2 berdasarkan fitur yang dialokasikan pada iterasi ke-2. Tabel 3.20 berikut merupakan alokasi data kluster 3 pada masing-masing fitur.

Tabel 3.20 Alokasi data kluster 3 iterasi 2 K-Means

No	K3F1	K3F2	K3F3	K3F4	K3F5
1					
2					
3					
4					
5	0.166569	1	0.146154	0.444444	0.083333
6					
7	0.160913	0.995859	0.253846	0.666667	0.5
8	0.000195	0.967288	0.033333	0.777778	0.333333
9	0	0.963147	0.035897	0.888889	0.333333
10	0.161498	0.993375	0.194872	1	0.3
Jumlah	0.489175	4.919669	0.664103	3.777778	1.55

Pada Tabel 3.20 menampilkan alokasi data kluster 3 berdasarkan fitur yang dialokasikan pada iterasi ke 2.

2. Menghitung pusat kluster dari data yang ada pada setiap kluster

$$C_{kj} = \frac{2.93817047}{3}$$

$$= 0.979390157$$

Sehingga diperoleh hasil yang ditampilkan pada Tabel 3.21 berikut:

Tabel 3.21 Pusat kluster iterasi 2 K-Means

Kel	Centroid F1	Centroid F2	Centroid F3	Centroid F4	Centroid F5
1	0.979390157	0.074396135	0.04786325	0.111111111	0.183333333
2	0.278525453	0.964389234	0.65769231	0.444444444	0.85
3	0.097834991	0.983933747	0.13282051	0.755555556	0.31

Pada Tabel 3.21 menampilkan *centroid* sebagai pusat kluster pada masing-masing fitur.

3. Menghitung Jarak data ke *centroid* dengan persamaan *Euclidian Distance*

$$\begin{aligned}
 d_{ik} &= \sqrt{(1 - 0.979390157)^2 + (0 - 0.074396135)^2 +} \\
 &\quad (0 - 0.047863248)^2 + \sqrt{(0 - 0.333333333)^2 + (0 - 0.4)^2} \\
 &= 0.976642005
 \end{aligned}$$

Sehingga diperoleh hasil yang ditampilkan pada Tabel 3.22 berikut:

Tabel 3.22 Jarak data ke Centroid iterasi 2 K-Means

No	Jarak ke C1	Jarak ke C2	Jarak Ke C3	Min	Kelompok Baru	Kelompok Sebelumnya
1	0.232824464	1.67426466	1.570551753	0.23282446	1	1
2	0.087008581	1.467703751	1.437026738	0.08700858	1	1
3	0.183446581	1.33054582	1.301022184	0.18344658	1	1
4	1.213758205	0.408082773	0.674335351	0.40808277	2	2
5	1.283819561	0.929113404	0.391571433	0.39157143	3	3
6	1.811448672	0.408082773	1.127899838	0.40808277	2	2
7	1.403690876	0.59143683	0.250538358	0.25053836	3	3
8	1.491051349	0.91943238	0.144036862	0.14403686	3	3
9	1.541643483	0.963681557	0.194216579	0.19421658	3	3
10	1.529318612	0.916449888	0.260471789	0.26047179	3	3
Jumlah	0.503279626	0.816165546	1.240835021			

Pada Tabel 3.22 menampilkan jarak data pada pusat kluster beserta penentuan jarak minimal data diantara ketiga kluster, sehingga dapat ditentukan pembaruan kluster baru dan kluster lama. Berikut merupakan perhitungan fungsi objektif.

F Baru = Jumlah C1+Jumlah C2+Jumlah C3

F Baru = 0.503279626 +0.816165546+ 1.240835021
= 2.560280194

Delta = 3.893563275 - 2.560280194
= 1.333283081

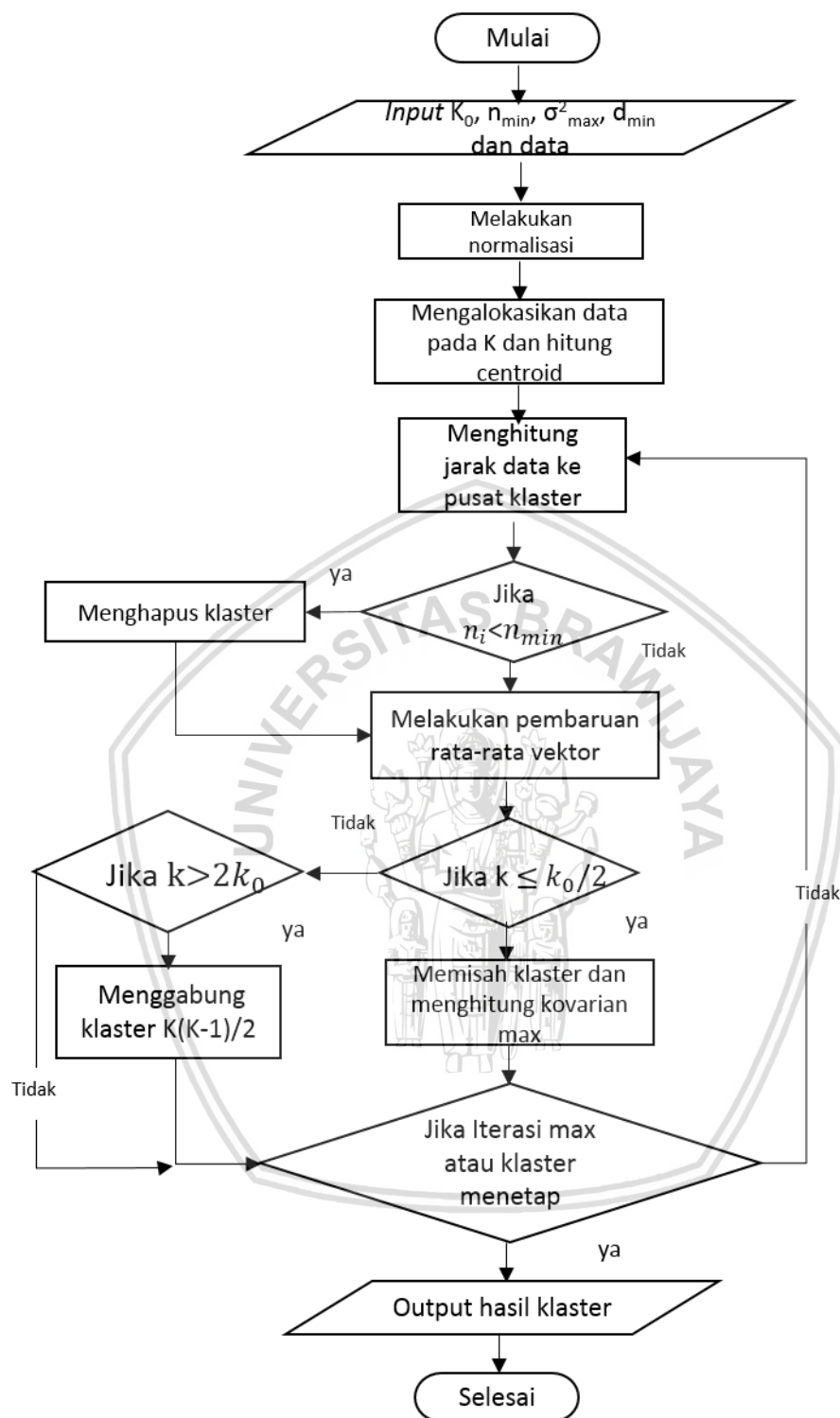
4. Pada iterasi Ke 2 tidak terdapat perpindahan data, sehingga Iterasi dihentikan.

3.4.3 Algoritme *Isodata*

Algoritme *Isodata* adalah algoritme kedua yang akan diterapkan dalam penelitian ini. Perancangan sistem pada Subbab 3.43, menggambarkan rancangan dengan menggunakan diagram alir untuk proses algoritme *Isodata*. Pada algoritme *Isodata* terdapat beberapa tahapan yang akan di lalui atau diproses. Pada tahapan yang pertama ketika sistem dijalankan, maka langkah yang pertama dilakukan pengguna adalah menentukan nilai K_0 , n_{min} , σ^2_{max} , d_{min} , dan masukan data titik panas atau data kejadian titik api yang akan di olah dengan algoritme *Isodata*.

Pada tahapan yang kedua sistem melakukan normalisasi dengan metode *min max* normalisasi. Setelah data dinormalisasi, tahapan yang ketiga adalah sistem mengalokasikan data pada setiap klaster secara acak dan menghitung pusat klaster pada masing-masing klaster yang telah dialokasikan. Pada tahapan ini, hasil data yang telah dinormalisasi memiliki klaster awal yang ditentukan secara acak. Pada tahapan keempat sistem menghitung jarak data ke seluruh pusat klaster. Pada tahapan ini data dapat berpindah klaster berdasarkan jarak terpendek pada salah satu pusat klaster. Pada tahapan yang kelima sistem menghilangkan klaster yang memiliki sedikit anggota dengan kondisi $n_j < n_{min}$ sebagai syarat untuk menghilangkan klaster. Pada tahapan keenam sistem melakukan pembaruan pusat klaster dengan cara menghitung nilai rata-rata masing-masing klaster.

Pada tahapan yang ketujuh sistem menyeleksi percabangan. Kondisi pertama jika $K \leq K_0/2$ maka sistem masuk pada tahapan ke-delapan untuk melakukan *split*. Sistem melakukan *split* dengan cara memisahkan klaster menjadi dua bagian yang memenuhi kondisi $\sigma^2_m > \sigma^2_{max}$ dan $n_j > 2n_{min}$ sebagai ketentuan dalam melakukan *split*. Pada kondisi kedua jika $K > 2K_0$ maka sistem masuk pada tahapan kesembilan untuk melakukan *merge*. Sistem melakukan *merge* dengan cara menggabungkan dua klaster yang memiliki jarak kurang dari d_{min} sebagai ketentuan dalam melakukan *merge*. Jarak masing-masing klaster dihitung berdasarkan jarak antar pusat klaster dengan menggunakan persamaan *Euclidean Distance*. Dua klaster yang digabungkan pusat klaster baru sehingga pada tahapan ini dilakukan pembaruan anggota klaster. Tahapan kesepuluh merupakan tahapan terakhir yaitu sistem menyeleksi percabangan dengan kondisi jika iterasi maksimal telah terpenuhi atau tidak terdapat data yang berpindah klaster, maka iterasi berhenti dan hasil akhir klasterisasi data kejadian titik api untuk wilayah Sumatera pada tahun 2001 hingga 2014 dapat ditetapkan, dengan kondisi lain sistem melakukan iterasi di mulai dari langkah keempat. Gambar 3.5 merupakan diagram alir algoritme *Isodata*.



Gambar 3.5 Diagram alir Algoritme *Isodata*.

3.4.4 Manualisasi Algoritme *Isodata*

Manualisasi algoritme *Isodata* merupakan manualisasi metode kedua yang digunakan pada studi kasus klasterisasi data kejadian titik api. Dalam tahapan manualisasi, data yang digunakan adalah sampel data titik kejadian titik api sebanyak sepuluh. Tabel 3.23 merupakan data sampel yang digunakan dalam proses manualisasi algoritme *Isodata*.

Tabel 3.23 Data sampel *Isodata*

No	Latitude	Longitude	Brigness	Acq_Date	Convidence
1	0.58	101.981	311.3	12/10/2005	35
2	0.568	102.16	312.5	12/11/2005	51
3	0.275	102.341	315.7	12/12/2005	52
4	-2.517	104.245	323.6	12/13/2005	77
5	-3.693	104.396	317	12/14/2005	40
6	-3.721	104.375	350.3	12/15/2005	95
7	-3.722	104.386	321.2	12/16/2005	65
8	-4.546	104.317	312.6	12/17/2005	55
9	-4.547	104.307	312.7	12/18/2005	55
10	-3.719	104.38	318.9	12/19/2005	53

Sebelum melakukan perhitungan manualisasi, data sampel pada Tabel 3.23 akan dilakukan normalisasi terlebih dahulu sehingga data menjadi data *range* dengan interval antara 0 sampai 1. Berikut merupakan perhitungan untuk mendapatkan nilai normalisasi.

$$\begin{aligned}\text{Normalisasi} &= \frac{0.58 - (-4.547)}{0.58 + 4.547} \\ &= 1\end{aligned}$$

Data yang dihitung pada persamaan min max normalisasi di atas merupakan data pada baris pertama kolom pertama sebagai sampel untuk menghitung nilai normalisasi dengan menggunakan *min max* normalisasi. Nilai data yang digunakan adalah $x = 0.58$ yang menghasilkan nilai normalisasi = 1. Pada Tabel 3.24 menjabarkan seluruh hasil normalisasi data sampel kejadian titik api yang digunakan.

Tabel 3.24 Normalisasi data sampel *Isodata*

No	F1	F2	F3	F4	F5
1	1	0	0	0	0
2	0.99765945	0.074120083	0.03076923	0.11111111	0.266666667
3	0.94051102	0.149068323	0.11282051	0.22222222	0.283333333
4	0.395943047	0.93747412	0.31538462	0.33333333	0.7
5	0.166569144	1	0.14615385	0.44444444	0.083333333
6	0.16110786	0.991304348	1	0.55555556	1
7	0.160912815	0.995859213	0.25384615	0.66666667	0.5
8	0.000195046	0.967287785	0.03333333	0.77777778	0.333333333
9	0	0.963146998	0.03589744	0.88888889	0.333333333
10	0.161497952	0.993374741	0.19487179	1	0.3

Setelah melakukan normalisasi data sampel, langkah pertama yang dilakukan adalah melakukan perhitungan manual dengan algoritme *Isodata*. Berikut merupakan proses perhitungan manual algoritme *Isodata*.

1. Menentukan Nilai $K_0 = 3$
2. Mengalokasikan data kedalam kluster secara acak. Tabel 3.25 merupakan alokasi data kedalam kluster.

Tabel 3.25 Alokasi data pada K *Isodata*

No	F1	F2	F3	F4	F5	K1	K2	K3
1	1	0	0	0	0	*		
2	0.99765945	0.074120083	0.03076923	0.11111111	0.266666667		*	
3	0.94051102	0.149068323	0.11282051	0.22222222	0.283333333		*	
4	0.395943047	0.93747412	0.31538462	0.33333333	0.7	*		
5	0.166569144	1	0.14615385	0.44444444	0.083333333		*	
6	0.16110786	0.991304348	1	0.55555556	1			*
7	0.160912815	0.995859213	0.25384615	0.66666667	0.5	*		
8	0.000195046	0.967287785	0.03333333	0.77777778	0.333333333			*
9	0	0.963146998	0.03589744	0.88888889	0.333333333		*	
10	0.161497952	0.993374741	0.19487179	1	0.3		*	
Jumlah						3	5	2

Pada Tabel 3.25 alokasi kluster setiap baris data ditandai dengan tanda (*) pada kolom K1, K2 dan K3. Jumlah data yang dialokasikan pada K1 sebagai kluster 1 adalah 3, jumlah data yang dialokasikan pada K2 sebagai kluster 2 adalah 3 dan jumlah data yang dialokasikan pada K3 sebagai kluster 3 adalah 2. Tabel 3.26 merupakan alokasi data ke dalam kluster 1 berdasarkan masing-masing fitur.

Tabel 3.26 Alokasi data kluster 1 *Isodata*

No	K1F1	K1F2	K1F3	K1F4	K1F5
1	1	0	0	0	0
2					
3					
4	0.39594305	0.93747412	0.315384615	0.333333	0.7
5					
6					
7	0.16091281	0.995859213	0.253846154	0.666667	0.5
8					
9					
10					
Jumlah	1.55685586	1.933333333	0.569230769	1	1.2

Tabel 3.26 merupakan tabel alokasi data kluster 1 yang menjabarkan jumlah nilai data yang dihitung berdasarkan masing-masing fitur. Tabel 3.27 merupakan alokasi data ke dalam kluster 2 berdasarkan masing-masing fitur.

Tabel 3.27 Alokasi data Kluster 2 *Isodata*

No	K1F1	K2F2	K2F3	K2F4	K2F5
1					
2	0.99765945	0.07412	0.030769	0.111111	0.266667
3	0.94051102	0.149068	0.112821	0.222222	0.283333
4					
5	0.16656914	1	0.146154	0.444444	0.083333
6					
7					
8					
9	0	0.963147	0.035897	0.888889	0.333333
10	0.16149795	0.993375	0.194872	1	0.3
Jumlah	2.26623757	3.17971	0.520513	2.666667	1.266667

Tabel 3.27 merupakan tabel alokasi data kluster 2 yang menjabarkan jumlah nilai data yang dihitung per fitur. Tabel 3.28 merupakan alokasi data ke dalam kluster 3 berdasarkan masing-masing fitur.

Tabel 3.28 Alokasi data kluster 3 Isodata

No	K3F1	K3F2	K3F3	K3F4	K3F5
1					
2					
3					
4					
5					
6	0.161108	0.991304	1	0.555556	1
7					
8	0.000195	0.967288	0.033333	0.777778	0.333333
9					
10					
Jumlah	0.161303	1.958592	1.033333	1.333333	1.333333

Tabel 3.28 merupakan tabel alokasi data kluster 3 yang menjabarkan jumlah nilai data yang dihitung per-fitur. Setelah mengalokasikan data ke dalam kluster berdasarkan masing-masing fitur maka langkah selanjutnya menghitung rata-rata x pada masing-masing fitur sebagai pusat kluster. Berikut merupakan perhitungan untuk mendapatkan nilai rata-rata sebagai pusat kluster atau *centroid*.

$$C_{kj} = \frac{1.55685586}{3}$$

$$= 0.518951954$$

Data yang dihitung diatas merupakan data pada K1F1 sebagai sampel untuk menghitung nilai *centroid*. Nilai data yang digunakan adalah 1.55685586 yang menghasilkan nilai *centroid* = 0.518951954. Pada Tabel 3.29 menjabarkan seluruh hasil normalisasi data sampel kejadian titik api yang digunakan.

Tabel 3.29 Centroid Isodata

kel	Centroid F1	Centroid F2	Centroid F3	Centroid F4	Centroid F5
1	0.518951954	0.644444444	0.18974359	0.33333333	0.4
2	0.453247513	0.635942029	0.10410256	0.53333333	0.25333333
3	0.080651453	0.979296066	0.51666667	0.66666667	0.66666667

Setelah mendapatkan pusat kluster pada Tabel 3.29, maka selanjutnya menghitung jarak setiap data ke pusat kluster dan mengalokasikan data ke dalam kluster berdasarkan jarak terpendak pada masing-masing pusat kluster. Berikut merupakan hasil dari perhitungan jarak data ke *centroid* terdekat yang ditampilkan pada Tabel 3.30 dengan menggunakan persamaan *Euclidean Distance*.

$$d_{ik} = \sqrt{(1 - 0.518951954)^2 + (0 - 0.644444444)^2 + (0 - 0.18974359)^2 + \sqrt{(0 - 0.33333333)^2 + (0 - 0.4)^2}}$$

$$= 0.976642005$$

Tabel 3.30 Jarak data Ke Centroid Isodata

No	Jarak ke C1	Jarak Ke C2	Jarak Ke C3	Min	Kelompok Baru	Kelompok Sebelumnya
1	0.976642005	1.030931672	1.72048129	0.976642	1	1
2	0.804278612	0.89210729	1.537850942	0.80427861	1	2
3	0.674524646	0.756464049	1.391475334	0.67452465	1	2
4	0.454734262	0.615106999	0.5038795	0.45473426	1	1
5	0.604249055	0.503281471	0.731266536	0.50328147	2	2
6	1.14641282	1.253899959	0.603063291	0.60306329	3	3
7	0.613925578	0.562179082	0.321820794	0.32182079	3	1
8	0.774452083	0.621456288	0.603063291	0.60306329	3	3
9	0.841211927	0.67079661	0.63118803	0.63118803	3	2
10	0.839044968	0.664131432	0.59652482	0.59652482	3	2

3. Menghilangkan klaster yang hanya memiliki sedikit anggota yaitu jika $n_j < n_{min}$ maka klaster dihilangkan dan memindahkan data pada klaster lain.

$$n_{min} = 2$$

Tabel 3.31 merupakan tabel status klaster.

Tabel 3.31 Status klaster Isodata

klaster	Jumlah	Status Klaster
1	4	Tetap
2	1	Hilangkan
3	5	Tetap

4. Melakukan pembaruan nilai rata-rata vektor pada klaster $\omega_j (j = 1, \dots, K)$

Tabel 3.32 Pembaruan klaster Isodata

No	F1	F2	F3	F4	F5	K1	K3
1	1	0	0	0	0	*	
2	0.99765945	0.074120083	0.03076923	0.11111111	0.266666667	*	
3	0.94051102	0.149068323	0.11282051	0.22222222	0.283333333	*	
4	0.395943047	0.93747412	0.31538462	0.33333333	0.7	*	
5	0.166569144	1	0.14615385	0.44444444	0.083333333	*	
6	0.16110786	0.991304348	1	0.55555556	1		*
7	0.160912815	0.995859213	0.25384615	0.66666667	0.5		*
8	0.000195046	0.967287785	0.03333333	0.77777778	0.333333333		*
9	0	0.963146998	0.03589744	0.88888889	0.333333333		*
10	0.161497952	0.993374741	0.19487179	1	0.3		*
Jumlah						5	5

Pada Tabel 3.32 menjabarkan pengalokasian data pada klaster yang terdiri dari dua klaster yaitu K1 dan K3. Pada Tabel 3.31 klaster K2 dihilangkan karena jumlah anggota kurang dari jumlah minimum klaster yang telah ditentukan. Tabel 3.33 merupakan pembaruan data pada klaster 1.

Tabel 3.33 Pembaruan data klaster 1 *Isodata*

No	K1F1	K1F2	K1F3	K1F4	K1F5
1	1	0	0	0	0
2	0.99765945	0.07412008	0.030769231	0.111111111	0.266667
3	0.94051102	0.14906832	0.112820513	0.222222222	0.283333
4	0.395943047	0.93747412	0.315384615	0.333333333	0.7
5	0.166569144	1	0.146153846	0.444444444	0.083333
6					
7					
8					
9					
10					
Jumlah	3.50068266	2.16066253	0.605128205	1.111111111	1.333333

Pada Tabel 3.33 menjabarkan alokasi data pada klaster 1. Tabel 3.34 merupakan pembaruan data pada klaster 3.

Tabel 3.34 Pembaruan data klaster 3 *Isodata*

No	K3F1	K3F2	K3F3	K3F4	K3F5
1					
2					
3					
4					
5					
6	0.161108	0.99130435	1	0.555556	1
7	0.160913	0.99585921	0.253846	0.666667	0.5
8	0.000195	0.96728778	0.033333	0.777778	0.333333
9	0	0.963147	0.035897	0.888889	0.333333
10	0.161498	0.99337474	0.194872	1	0.3
Jumlah	0.483714	4.91097308	1.517949	3.888889	2.466667

Pada Tabel 3.34 menjabarkan alokasi data pada klaster 3. Setelah melakukan alokasi data pada seluruh klaster, maka tahapan selanjutnya adalah melakukan pembaruan pusat klaster.

$$C_{kj} = \frac{3.50068266}{5}$$

$$= 0.700136532$$

Data yang digunakan dalam perhitungan diatas merupakan data pada kluster 1 fitur 1. Tabel 3.35 merupakan hasil pembaruan seluruh kluster.

Tabel 3.35 Pembaruan pusat kluster (Isodata)

kel	<i>Centroid F1</i>	<i>Centroid F2</i>	<i>Centroid F3</i>	<i>Centroid F4</i>	<i>Centroid F5</i>
1	0.700136532	0.432132505	0.12102564	0.22222222	0.266666667
3	0.096742735	0.982194617	0.30358974	0.77777778	0.493333333

5. Jika $K \leq K_0 / 2$, maka lakukan langkah 6 untuk melakukan pemisahan kluster dan jika $K > 2K_0$ maka lakukan langkah 7 untuk melakukan pengabungan kluster

$$K = 2$$

$$K_0 = 3$$

$$K_0/2 = 3/2$$

$$= 1.5$$

$$2K_0 = 2 * 3$$

$$= 6$$

Berdasarkan hasil yang didapat pada langkah lima, proses langsung melakukan langkah 8 dikarenakan hasil dari langkah lima tidak memenuhi syarat $K \leq K_0 / 2$ atau $K > 2K_0$.

6. Kembali pada langkah 2 apabila jumlah iterasi maksimum belum tercapai, Jika iterasi maksimum tercapai, maka berhenti melakukan iterasi.

Iterasi Ke 1

1. Menghitung jarak tiap data ke pusat kluster dan mengalokasikan data ke dalam kluster berdasarkan jarak terpendak pada pusat kluster. Tabel 3.6 berikut merupakan hasil dari perhitungan jarak data ke *centroid* terdekat dengan menggunakan persamaan *Euclidean Distance*.

Tabel 3.36 Jarak Data Ke *Centroid* Iterasi 1 Isodata

No	Jarak Ke C1	Jarak Ke C3	Min	Kelompok Baru	Kelompok Lama
1	0.641714605	1.649564408	0.641714605	1	1
2	0.487016098	1.485430423	0.487016098	1	1
3	0.371820289	1.339842982	0.371820289	1	1
4	0.765377483	0.576109229	0.576109229	3	1
5	0.831139521	0.556048498	0.556048498	3	1
6	1.422923003	0.891799826	0.891799826	3	3
7	0.937105446	0.138452405	0.138452405	3	3
8	1.04741855	0.328910622	0.328910622	3	3
9	1.108288435	0.345437142	0.345437142	3	3
10	1.103008162	0.320777679	0.320777679	3	3

- Menghilangkan kluster yang hanya memiliki sedikit anggota yaitu jika $n_j < n_{min}$ maka kluster dihilangkan dan memindahkan data pada kluster lain.

$$n_{min} = 2$$

Tabel 3.37 menampilkan status Kluster

Tabel 3.37 Status kluster iterasi 1 Isodata

kel	Jumlah	Status Kluster
1	3	Tetap
3	7	Tetap

Pada Tabel 3.37 menampilkan status kluster apakah terdapat perubahan atau tidak.

- Melakukan pembaruan nilai rata-rata vektor pada kluster $\omega_j (j = 1, \dots, K)$

Tabel 3.38 Pembaruan Kluster Iterasi 1 Isodata

No	F1	F2	F3	F4	F5	K1	K3
1	1	0	0	0	0	*	
2	0.99765945	0.074120083	0.03076923	0.11111111	0.26666667	*	
3	0.94051102	0.149068323	0.11282051	0.22222222	0.28333333	*	
4	0.395943047	0.93747412	0.31538462	0.33333333	0.7		*
5	0.166569144	1	0.14615385	0.44444444	0.08333333		*
6	0.16110786	0.991304348	1	0.55555556	1		*
7	0.160912815	0.995859213	0.25384615	0.66666667	0.5		*
8	0.000195046	0.967287785	0.03333333	0.77777778	0.33333333		*
9	0	0.963146998	0.03589744	0.88888889	0.33333333		*
10	0.161497952	0.993374741	0.19487179	1	0.3		*
Jumlah						3	7

Pada Tabel 3.38 menampilkan pembaruan kluster pada iterasi 1. Tabel 3.39 merupakan pembaruan data pada kluster 1.

Tabel 3.39 Pembaruan data kluster 1 Isodata

No	K1F1	K1F2	K1F3	K1F4	K1F5
1	1	0	0	0	0
2	0.99765945	0.07412008	0.030769231	0.11111111	0.266667
3	0.94051102	0.14906832	0.112820513	0.22222222	0.283333
4					
5					
6					
7					
8					
9					

Tabel 3.39 Pembaruan data kluster 1 (lanjutan)

No	K1F1	K1F2	K1F3	K1F4	K1F5
10					
Jumlah	2.93817047	0.22318841	0.143589744	0.333333333	0.55

Pada Tabel 3.39 menampilkan pembaruan data pada kluster 1. Tabel 3.40 merupakan pembaruan data pada kluster 3.

Tabel 3.40 Pembaruan data kluster 3 iterasi 1 *Isodata*

No	K3F1	K3F2	K3F3	K3F4	K3F5
1					
2					
3					
4	0.395943	0.93747412	0.315385	0.333333	0.7
5	0.166569	1	0.146154	0.444444	0.083333
6	0.161108	0.99130435	1	0.555556	1
7	0.160913	0.99585921	0.253846	0.666667	0.5
8	0.000195	0.96728778	0.033333	0.777778	0.333333
9	0	0.963147	0.035897	0.888889	0.333333
10	0.161498	0.99337474	0.194872	1	0.3
Jumlah	1.046226	6.8484472	1.979487	4.666667	3.25

Pada Tabel 3.40 menampilkan pembaruan data kluster 3 pada iterasi 1. Tabel 3.41 menyajikan hasil perhitungan pembaruan rata-rata vektor sebagai pusat kluster 1 dan 3.

Tabel 3.41 Pembaharuan Pusat Kluster Iterasi 1 (*Isodata*)

Kel	Centroid F1	Centroid F2	Centroid F3	Centroid F4	Centroid F5
1	0.979390157	0.074396135	0.04786325	0.11111111	0.18333333
3	0.149460838	0.978349601	0.28278388	0.66666667	0.464285714

4. Jika $K \leq K_0/2$, maka lakukan langkah 6 untuk melakukan pemisahan kluster dan jika $K > 2K_0$ maka lakukan langkah 7 untuk melakukan penggabungan kluster

$$K = 2$$

$$K_0 = 3$$

$$K_0/2 = 3/2$$

$$= 1.5$$

$$2K_0 = 2 * 3$$

$$= 6$$

Berdasarkan hasil yang didapat pada langkah lima, proses langsung melakukan langkah 8 dikarenakan hasil dari langkah lima tidak memenuhi syarat $K \leq K_0/2$ atau $K > 2K_0$.

1. Kembali pada langkah 2 apabila jumlah iterasi maksimum belum tercapai, Jika iterasi maksimum tercapai, maka berhenti melakukan iterasi.

Iterasi ke 2

1. Menghitung jarak data ke pusat kluster dengan menggunakan persamaan *Euclidean Distance*. Tabel 3.42 menyajikan hasil perhitungan jarak data ke pusat kluster dan alokasi data pada pusat kluster terdekat.

Tabel 3.42 Jarak data ke centroid iterasi 2 *Isodata*

No	Jarak Ke C1	Jarak Ke C3	Min	Kelompok Baru	Kelompok Lama
1	0.232824464	1.555814	0.232824464	1	1
2	0.087008581	1.395807492	0.087008581	1	1
3	0.183446581	1.254045455	0.183446581	1	1
4	1.213758205	0.479749338	0.479749338	3	3
5	1.283819561	0.462532846	0.462532846	3	3
6	1.811448672	0.902240496	0.902240496	3	3
7	1.403690876	0.050503828	0.050503828	3	3
8	1.491051349	0.337820103	0.337820103	3	3
9	1.541643483	0.387367811	0.387367811	3	3
10	1.529318612	0.382361203	0.382361203	3	3

2. Berdasarkan Tabel 3.42 hasil pengelompokan kluster baru yang diperoleh sama dengan kluster lama, sehingga dalam hal ini tidak ada perpindahan kluster yang terjadi pada tahap ini.
3. Melakukan pembaruan nilai rata-rata vektor pada kluster $\omega_j (j = 1, \dots, K)$

Tabel 3.43 Pembaruan kluster iterasi 2 *Isodata*

No	F1	F2	F3	F4	F5	K1	K3
1	1	0	0	0	0	*	
2	0.99765945	0.074120083	0.030769231	0.111111111	0.266666667	*	
3	0.94051102	0.149068323	0.112820513	0.222222222	0.283333333	*	
4	0.395943047	0.93747412	0.315384615	0.333333333	0.7		*
5	0.166569144	1	0.146153846	0.444444444	0.083333333		*
6	0.16110786	0.991304348	1	0.555555556	1		*
7	0.160912815	0.995859213	0.253846154	0.666666667	0.5		*
8	0.000195046	0.967287785	0.033333333	0.777777778	0.333333333		*
9	0	0.963146998	0.035897436	0.888888889	0.333333333		*
10	0.161497952	0.993374741	0.194871795	1	0.3		*

Tabel 3.43 pembaruan kluster iterasi 2 *Isodata* (lanjutan)

No	F1	F2	F3	F4	F5	K1	K3
10	0.161497952	0.993374741	0.194871795	1	0.3		*
Jumlah						3	7

Pada Tabel 4.43 menampilkan pembaruan kluster pada iterasi 2. Tabel 3.44 berikut merupakan pembaruan data pada kluster 1.

Tabel 3.44 Pembaruan kluster 1 iterasi 2 *Isodata*

K1F1	K1F2	K1F3	K1F4	K1F5
1	0	0	0	0
0.997659	0.07412008	0.030769231	0.111111	0.266667
0.940511	0.14906832	0.112820513	0.222222	0.283333
2.93817	0.22318841	0.143589744	0.333333	0.55

Pada Tabel 3.44 menampilkan pembaruan data pada kluster 1. Tabel 3.45 merupakan pembaruan data pada kluster 3.

Tabel 3.45 Pambaruan kluster 3 iterasi 2 *Isodata*

K3F1	K3F2	K3F3	K3F4	K3F5
0.395943	0.93747412	0.315385	0.333333	0.7
0.166569	1	0.146154	0.444444	0.083333
0.161108	0.99130435	1	0.555556	1
0.160913	0.99585921	0.253846	0.666667	0.5
0.000195	0.96728778	0.033333	0.777778	0.333333
0	0.963147	0.035897	0.888889	0.333333
0.161498	0.99337474	0.194872	1	0.3
1.046226	6.8484472	1.979487	4.666667	3.25

Pada Tabel 3.45 menampilkan pembaruan data kluster 3 pada interasi 2. Tabel 3.46 menyajikan hasil perhitungan rata-rata vektor sebagai pembaruan pusat kluster.

Tabel 3.46 Pembaruan Pusat Klaster Iterasi 2 *Isodata*

Kel	Centroid F1	Centroid F2	Centroid F3	Centroid F4	Centroid F5
1	0.979390157	0.074396135	0.04786325	0.111111111	0.183333333
3	0.149460838	0.978349601	0.28278388	0.66666667	0.464285714

4. Jika $K \leq K_0/2$, maka lakukan langkah 6 untuk melakukan pemisahan Klaster dan jika $K > 2K_0$ maka lakukan langkah 7 untuk melakukan penggabungan klaster

$$K = 2$$

$$K_0 = 3$$

$$K_0/2 = 3/2$$

$$= 1.5$$

$$2K_0 = 2 * 3$$

$$= 6$$

Berdasarkan hasil yang didapat pada langkah lima, proses langsung melakukan langkah 8 dikarenakan hasil dari langkah ke-lima tidak memenuhi syarat $K \leq K_0/2$ atau $K > 2K_0$. pada tahapan ini iterasi dihentikan.

3.5 Implementasi

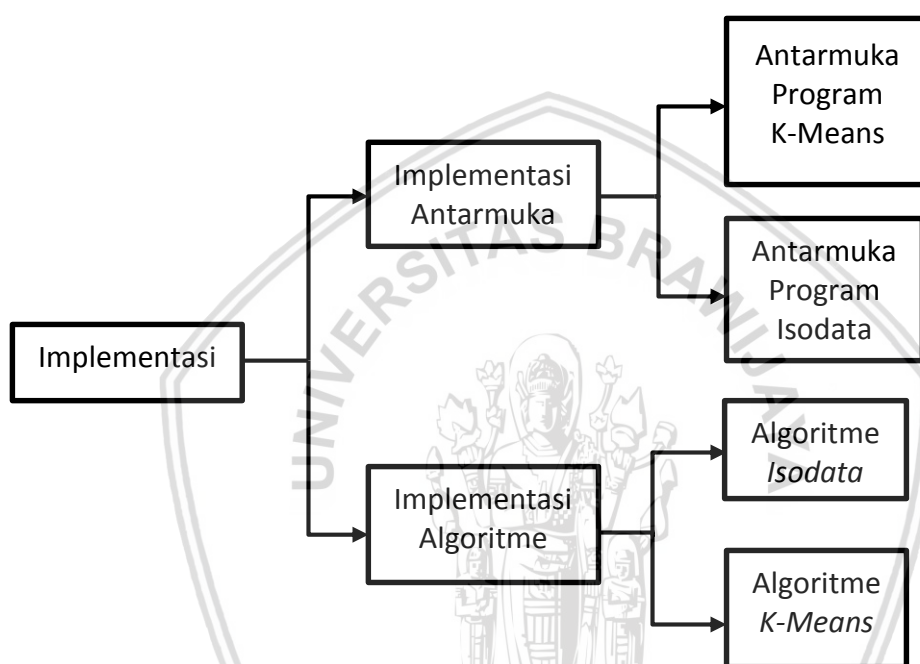
Implementasi merupakan tahapan untuk menerapkan hasil dari perancangan yang telah dibuat pada tahap sebelumnya. Dalam penelitian ini implementasi dilakukan dengan cara membuat program yang menerapkan algoritme *K-Means* dan *Isodata* pada studi kasus klasterisasi data kejadian titik api wilayah Sumatera pada tahun 2001 hingga 2014. Bahasa pemrograman yang digunakan dalam penelitian ini, adalah bahasa pemrograman java.

3.6 Pengujian dan Analisis

Pengujian dan analisis merupakan tahapan untuk melakukan evaluasi hasil dari implementasi yang telah dilakukan. Dalam penelitian ini, hasil klaster dari penerapan algoritme *K-Means* dan *Isodata* di uji berdasarkan kualitas klaster yang dihasilkan. Metode yang digunakan untuk mengukur kualitas klaster dalam penelitian ini, adalah *Silhouette Coefficient* (SC). Hasil dari pengujian yang dilakukan dengan metode *Silhouette Coefficient* akan di jadikan sebagai acuan dalam melakukan analisis perbandingan antara metode *K-Means* dan *Isodata* pada studi kasus klasterisasi data kejadian titik api di wilayah sumatera pada tahun 2001 hingga 2014.

BAB 4 IMPLEMENTASI

Pada bab ini membahas tentang penerapan dari hasil perancangan yang telah di bahas pada Bab 3. Pada Sub bab 3.4 membahas tentang perancangan sistem dengan menggunakan algoritme *K-Means* dan *Isodata*. Hasil Perancangan sistem algoritme *K-Means* dan *Isodata* diimplementasikan pada program dengan bahasa pemrograman java. Gambar 4.1. menggambarkan cakupan implementasi yang dilakukan dalam penelitian ini.



Gambar 4.1 Alur implementasi

4.1 Implementasi antarmuka

Pada bahasan sub bab implementasi antarmuka membahas tentang antarmuka atau tampilan program yang dibangun berdasarkan perancangan sistem. Program yang dibuat terbagi menjadi dua bagian, diantaranya program *K-Means* dan program *Isodata*.

4.1.1 Antarmuka program *K-Means*

Pada subbab Antarmuka Program *K-Means* membahas hal yang terkait dengan tampilan program dengan metode *K-Means*.

4.1.2 Antarmuka awal *K-Means*

Antarmuka tampilan awal *K-Means* adalah tampilan awal ketika program *K-Means* dijalankan. Gambar 4.2 menampilkan antarmuka tampilan awal program dengan metode *K-Means*.

**K-Means untuk Data Kejadian Titik Api Wilayah Sumatera
Pada Tahun 2001 Hingga 2014**

K Hasil

Treshold

Data

Representasi Data

Title 1	Title 2	Title 3	Title 4

Normalisasi

Title 1	Title 2	Title 3	Title 4

Gambar 4.2 Tampilan awal program *K-Means*

Pada Gambar 4.2 menampilkan tampilan awal program *K-Means* untuk data kejadian titik api wilayah Sumatera pada Tahun 2001 hingga 2014 ketika program dijalankan. Pada gambar 4.2 tampak kolom tabel dan variabel masih kosong.

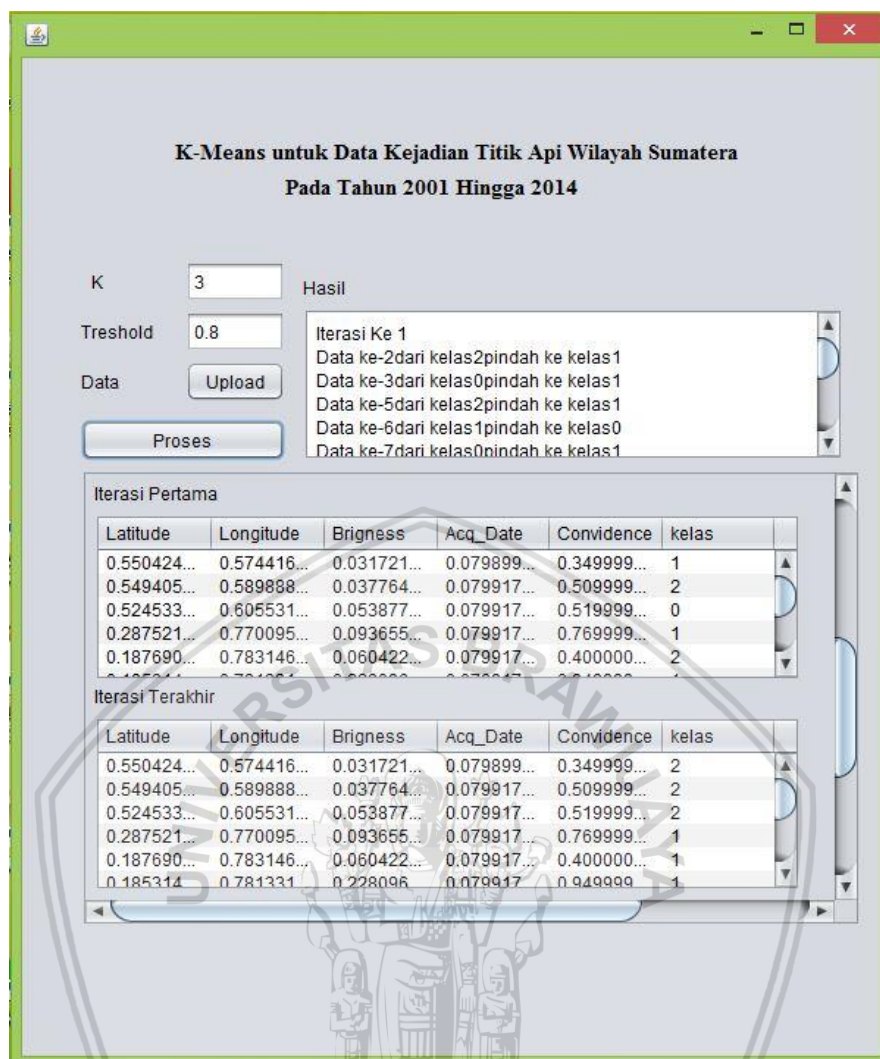
4.1.3 Antarmuka input variabel *K-Means*

Antarmuka *input* variabel *K-Means* merupakan tampilan *input* variabel program *K-Means*. Gambar 4.3 merupakan tampilan *input* variabel program *K-Means*.



4.1.4 Antarmuka hasil *scrool* atas *K-Means*

46

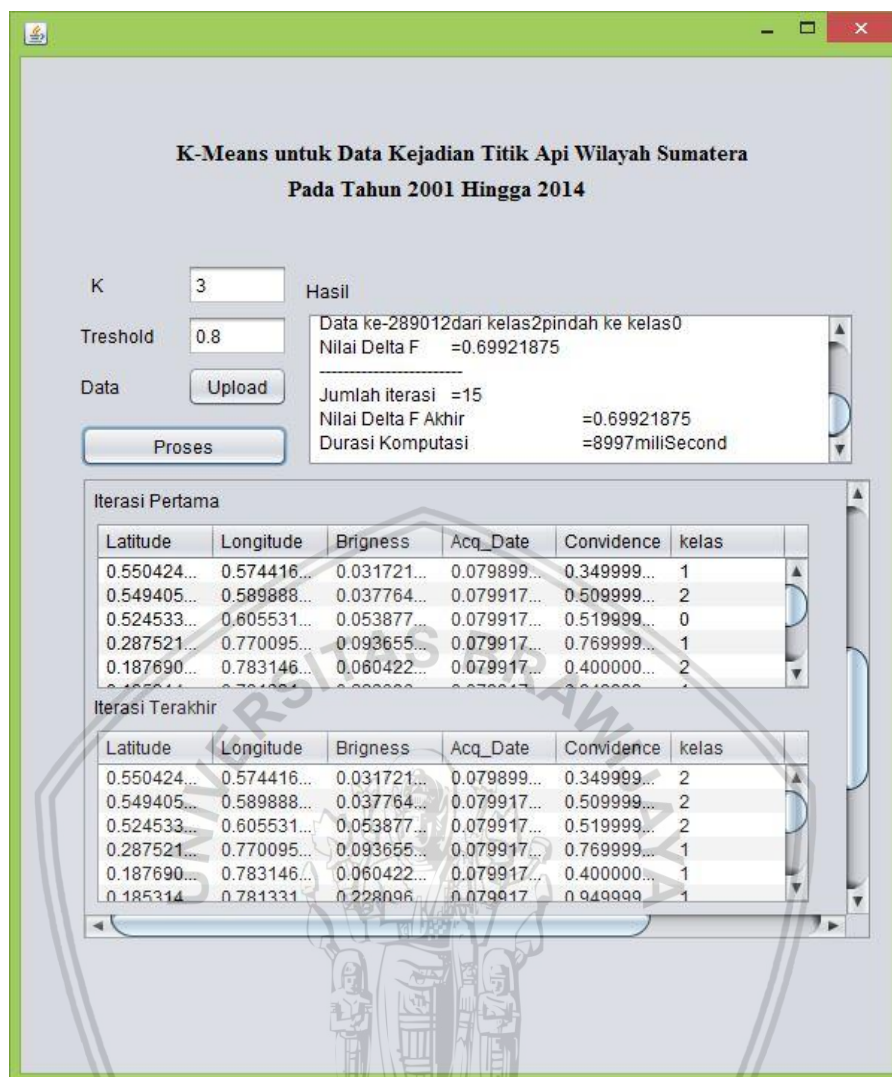


Gambar 4.4 Antarmuka hasil *scroll* atas *K-Means*

Pada Gambar 4.4 terdapat tabel normalisasi yang merupakan hasil dari normalisasi data yang telah di unggah sebelumnya, sementara pada tampilan kolom hasil menampilkan proses perpindahan klaster.

4.1.5 Antarmuka hasil *scroll* bawah *K-Means*

Antarmuka hasil *scroll* bawah *K-Means* merupakan tampilan hasil dari proses yang telah dijalankan dengan algoritme *K-Means*. Gambar 4.5 merupakan antarmuka hasil bagian bawah program *K-Means*.



Gambar 4.5 Antarmuka hasil scroll bawah K-Means

Pada Gambar 4.5 menampilkan hasil dari proses algoritme *K-Means* berdasarkan variabel yang telah dimasukkan sebelumnya. Gambar 4.5 menampilkan tabel hasil klaster dari iterasi pertama dan hasil klaster dari iterasi terakhir, sementara pada kolom hasil menampilkan nilai jumlah iterasi, nilai *delta* akhir dan durasi komputasi.

4.1.6 Antarmuka program *Isodata*

Pada subbab Antarmuka program *Isodata* membahas hal yang terkait dengan tampilan program *Isodata*.

4.1.7 Antarmuka awal *Isodata*

Antarmuka awal *Isodata* adalah tampilan awal ketika program *Isodata* dijalankan. Gambar 4.2 menampilkan antarmuka tampilan awal program *Isodata*.

**Isodata Untuk Klasterisasi Data Kejadian Titik Api Wilayah Sumatera
Pada Tahun 2001 Hingga 2014**

K0 Result

N min

Max Var

d min

Max Iteration

Data

Representasi Data

Title 1	Title 2	Title 3	Title 4

Normalisasi

Title 1	Title 2	Title 3	Title 4

Gambar 4.6 Antarmuka Tampilan Awal Isodata

Pada Gambar 4.6 menampilkan tampilan awal program *Isodata* ketika program dijalankan. Pada Gambar 4.6 tampak kolom tabel dan variabel masih kosong.

4.1.8 Antarmuka masukan variabel *Isodata*

Antarmuka tampilan input variabel *Isodata* merupakan tampilan masukan variabel program *Isodata*. Gambar 4.7 merupakan tampilan masukan variabel program *Isodata*.

Isodata Untuk Klasterisasi Data Kejadian Titik Api Wilayah Sumatera
Pada Tahun 2001 Hingga 2014

K0 Result

N min

Max Var

d min

Max Iteration

Data

Representasi Data

Latitude	Longitude	Brigness	Acq_Date	Convidence
0.58	101.981	311.3	2005-12-10	35
0.568	102.16	312.5	2005-12-13	51
0.275	102.341	315.7	2005-12-13	52
-2.517	104.245	323.6	2005-12-13	77
-3.693	104.396	317.0	2005-12-13	40
-3.721	104.375	350.3	2005-12-13	95
-3.722	104.386	321.2	2005-12-13	65

Normalisasi

Title 1	Title 2	Title 3	Title 4

Gambar 4.7 Antarmuka masukan variabel *Isodata*

Pada Gambar 4.7 tampilan program *Isodata* yang telah diisi dengan masukan variabel. Pada Gambar 4.7 terdapat lima variabel *input* yang akan diolah dengan algoritme *Isodata* yaitu K_0 , n_{min} , $Max Var$, d_{min} , $Max iteration$ dan data yang telah diunggah.

4.1.9 Antarmuka hasil *scroll* atas *Isodata*

Antarmuka hasil *scroll* atas *Isodata* merupakan tampilan hasil dari proses yang telah dijalankan dengan algoritme *Isodata* setelah pengguna mengklik tombol proses. Gambar 4.8 merupakan tampilan antarmuka hasil bagian atas program *Isodata*.

**Isodata Untuk Klasterisasi Data Kejadian Titik Api Wilayah Sumatera
Pada Tahun 2001 Hingga 2014**

K0 Result

Iterasi Ke1
Iterasi Ke2
Iterasi Ke3
Iterasi Ke4
Iterasi Ke5

N min

Max Var

d min

Max Iteration

Data

Representasi Data

Latitude	Longitude	Brigness	Acq_Date	Convidence
0.58	101.981	311.3	2005-12-10	35
0.568	102.16	312.5	2005-12-13	51
0.275	102.341	315.7	2005-12-13	52
-2.517	104.245	323.6	2005-12-13	77
-3.693	104.396	317.0	2005-12-13	40
-3.721	104.375	350.3	2005-12-13	95
-3.722	104.386	321.2	2005-12-13	65

Normalisasi

Latitude	Longitude	Brigness	Acq_date	Convidence
0.55042445...	0.57441693...	0.03172199...	0.79904484...	0.34999999...
0.54940575...	0.58988803...	0.03776434...	0.79922747...	0.50999999...
0.52453309...	0.60553193...	0.05387720...	0.79922747...	0.51999998...
0.28752121...	0.77009540...	0.09365561...	0.79922747...	0.76999998...
0.18769098...	0.78314650...	0.06042296...	0.79922747...	0.40000000...
0.18531408...	0.78133112...	0.22809660...	0.79922747...	0.94999998...
0.18522910...	0.78228109...	0.08157105...	0.79922747...	0.61000007...

Gambar 4.8 Antaruka Hasil Scroll Atas Isodata

Pada Gambar 4.8 terdapat tabel hasil normalisasi data yang telah di unggah sebelumnya, sementara pada tampilan kolom hasil menampilkan proses perpindahan klaster.

4.1.10 Antarmuka hasil scroll bawah *Isodata*

Antarmuka hasil *Scroll* bawah *Isodata* merupakan tampilan hasil dari proses yang telah dijalankan dengan algoritme *Isodata* setelah pengguna mengklik tombol proses. Gambar 4.8 merupakan tampilan antarmuka hasil bagian bawah program *Isodata*.

Gambar 4.9 Antarmuka Scroll Bawah Isodata

Pada Gambar 4.9 menampilkan hasil dari proses algoritme *Isodata* berdasarkan variabel yang telah dimasukkan sebelumnya. Gambar 4.9 menampilkan tabel hasil klaster dari iterasi pertama dan hasil klaster dari iterasi terakhir, sementara pada kolom hasil menampilkan nilai jumlah iterasi, Jumlah data per klaster dan durasi komputasi.

4.2 Implementasi algoritme

Pada subbab implementasi algoritme membahas tentang penerapan algoritme *K-Means* dan *Isodata* dalam bahasa pemrograman java. Pada studi kasus ini algoritme *K-Means* dan *Isodata* diimplementasikan pada data kejadian titik api wilayah Sumatera pada tahun 2001 hingga 2014.

4.2.1 Implementasi algoritme *K-Means*

Dalam implementasi algortime *K-Means* terdapat beberapa proses yang akan dijalankan oleh sistem. Berikut merupakan bagian bagian implementasi dari algortime *K-Means*.

1. Deklarasi K , $threshold$ dan muatan data

Langkah pertama yang dilakukan pada proses algoritme *K-Means* adalah menentukan nilai K , *threshold* dan muatan data.

Algoritme 1 : input variabel K-Means	
1	private int k;
2	private float treshold;
3	private void muatData(String lokasi) throws
4	FileNotFoundException, IOException {
5	File csvFile = new File(lokasi);
6	
7	if (csvFile.exists()) {
8	try {
9	FileReader fr = new FileReader(csvFile);
10	BufferedReader br = new BufferedReader(fr);
11	String st;
12	int baris = 0;
13	a.clear();
14	while ((st = br.readLine()) != null) {
15	if (baris > 0) {
16	String[] data = st.split(",");
17	float latitude =
18	Float.parseFloat(data[1]);
19	float longitude =
20	Float.parseFloat(data[2]);
21	float brigness =
22	Float.parseFloat(data[3]);
23	Date date = Date.valueOf(data[6]);
24	int convidence =
25	Integer.parseInt(data[9]);
26	a.add(new KlasterKmeans(latitude,
27	longitude,
28	brigness, date, convidence));
29	}
30	baris++;
31	}
32	String header[] = {"Latitude", "Longitude",
33	"Brigness",
34	"Acq_Date", "Convidence"};
35	DefaultTableModel model = new
36	DefaultTableModel(null, header);
37	model.setColumnCount(5);
38	model.setRowCount(baris - 1);
39	for (int i = 0; i < (baris - 1); i++) {
40	model.setValueAt(a.get(i).latitude, i,
41	0);
42	model.setValueAt(a.get(i).longitude, i,
43	1);
44	model.setValueAt(a.get(i).brigness, i,
45	2);
46	model.setValueAt(a.get(i).acq_date, i,
47	3);
48	model.setValueAt(a.get(i).convidence,
49	i, 4);
50	}
51	jTable1.setModel(model);
52	} catch (Exception e) {
53	JOptionPane.showMessageDialog(null, "Error: " + e);
54	}
55	}else{
56	JOptionPane.showMessageDialog(null, "File does
57	not
58	does not
59	exist");
60	} }

Penjelasan Algoritme 1:

- Baris 1 dan 2 adalah deklarasi variabel *K* dan *threshold* dengan tipe data integer dan float.
- Baris 3 merupakan kode fungsi untuk memuat data.
- Baris 4 adaah kode eksepsi.
- Baris 5 berfungsi untuk menjalankan objek untuk file CSV.
- Baris 7-55 adalah kondisi ketika file csv tersedia sehingga program dapat membaca data dengan format csv. Pada proses ini data difilter berdasarkan fitur yang dibutuhkan. Pada tahapan ini fitur yang dipilih adalah latitude, longitude, brightness, acq_date, convidence.
- Baris 56-59 adalah kondisi lain apabila kondisi pada baris ke 7 tida terpenuhi.

2. Melakukan normalisasi data

Pada tahap kedua dalam proses ini adalah melakukan normalisasi data, dimana pada studi kasus ini menggunakan normalisasi min max. Algoritme 2 merupakan implementasi proses normalisasi dalam bahasa pemrograman java.

Algoritme 2 : metode normalisasi	
1	private void normalisasi(){
2	float maxLatitude = 0, minLatitude =1000;
3	float maxLongitude =0, minLongitude =1000;
4	float maxBrigness = 0, minBrigness= 1000;
5	float maxConvidence =0, minConvidence=1000;
6	float maxDate = 0, minDate = 1000;
7	float date =0;
8	for(int i =0; i<a.size(); i++){
9	if(a.get(i).latitude>maxLatitude){
10	maxLatitude=a.get(i).latitude;
11	}
12	if(a.get(i).latitude<minLatitude){
13	minLatitude=a.get(i).latitude;
14	}
15	if(a.get(i).longitude>maxLongitude){
16	maxLongitude=a.get(i).longitude;
17	}
18	if(a.get(i).longitude<minLongitude){
19	minLongitude = a.get(i).longitude;
20	}
21	if(a.get(i).brigness>maxBrigness){
22	maxBrigness=a.get(i).brigness;
23	}
24	if(a.get(i).brigness<minBrigness){
25	minBrigness=a.get(i).brigness;
26	}
27	if (a.get(i).convidence>maxConvidence){
28	maxConvidence=a.get(i).convidence;
29	}
30	if (a.get(i).convidence>maxConvidence){
31	maxConvidence=a.get(i).convidence;
32	}
33	if (a.get(i).convidence<minConvidence){

Algoritma 2 : metode normalisasi (lanjutan)	
34	minConvience=a.get(i).convience;
35	}
36	date = a.get(i).acq_date.getTime()/10000;
37	if(date>maxDate){
38	maxDate=date;
39	}
40	}
41	String
42	header[]={ "Latitude", "Longitude", "Brigness", "Acq_Date", "Convience
43	", "Kelas"};
44	DefaultTableModel model = new DefaultTableModel(null,
45	header);
46	model.setColumnCount(5);
47	model.setRowCount(a.size());
48	for(int i =0; i<(a.size()); i++){
49	data.add(new ArrayList<Double>());
50	data.get(i).add((double)((a.get(i).latitude-
51	minLatitude)/(maxLatitude-minLatitude));
52	data.get(i).add((double)((a.get(i).longitude-
53	minLongitude)/(maxLongitude-minLongitude));
54	data.get(i).add((double)((a.get(i).brigness-
55	minBrigness)/(maxBrigness-minBrigness));
56	
57	data.get(i).add((double)((a.get(i).acq_date.getTime()/100000-
58	minDate)/(maxDate-minDate));
59	data.get(i).add((double)((a.get(i).convience-
60	minConvience)/(maxConvience-minConvience));
61	model.setValueAt(data.get(i).get(0), i, 0);
62	model.setValueAt(data.get(i).get(0), i, 0);
63	model.setValueAt(data.get(i).get(1), i, 1);
64	model.setValueAt(data.get(i).get(2), i, 2);
65	
66	model.setValueAt(data.get(i).get(2), i, 2);
67	model.setValueAt(data.get(i).get(3), i, 3);
68	model.setValueAt(data.get(i).get(4), i, 4);
69	}
70	tbNormalisasi.setModel(model);
71	}

Penjelasan Algoritme 2:

- Baris 1 berfungsi untuk mendefinisikan metode normalisasi.
- Baris 2-7 berfungsi untuk mendeklarasikan variabel beserta nilainya.
- Baris 8-40 menjabarkan perulangan untuk melakukan pengecekan dan penentuan nilai min dan max pada masing-masing fitur.
- Baris 41-45 berfungsi untuk mencetak header data.
- Baris 44-47 berfungsi untuk mengatur baris dan kolom tabel.
- Baris 48-60 menjabarkan perulangan untuk menghitung hasil normalisasi data dengan rumus min max normalisasi untuk seluruh fitur.
- Baris 61-70 berfungsi untuk memanggil nilai hasil normalisasi sehingga dapat ditampilkan pada tabel normalisasi.

3. Mengalokasikan data pada kluster

Pada tahap ketiga proses yang dilakukan adalah mengalokasikan data pada kluster secara acak. Pada algoritme 3 menjabarkan implementasi algoritme untuk mengalokasi data secara acak.

Algoritme 3 : Acak kelas	
1	for(int i=0; i<a.size(); i++){
2	kelas[i][0]=0+(int) (Math.random()*k);
3	}

Penjelasan algoritme 3:

- Baris 1 menjabarkan perulangan untuk mengacak data.
- Baris 2 menjabarkan formulasi untuk mengacak data.

4. Menghitung Pusat Kluster

Pada tahap keempat langkah yang yang dilakukan adalah menghitung pusat kluster. Algoritme 4 menjabarkan implementasi algoitme untuk menghitung pusat kluster atau yang lebih dikenal dengan *centroid* masing-masing kluster.

Algoritme 4 : Menghitung pusat kluster	
1	for(int i=0; i<k; i++){
2	totalClusterX.add(new ArrayList<Integer>());
3	int tempTotalClusterX=0;
4	for(int j=0; j<data.size(); j++){
5	if(kelas[j][0]==i){
6	for(int l=0; l<data.get(i).size(); l++){
7	clusterCentroid[i][l] +=
8	data.get(j).get(l);
9	}
10	tempTotalClusterX++;
11	}
12	}
13	for(int j=0; j<clusterCentroid[i].length; j++){
14	clusterCentroid[i][j]/=(float)
15	tempTotalClusterX;
16	}
17	}

Penjelasan Algoritme 4:

- Baris 1 menjabarkan perulangan untuk menghitung pusat kluster sebanyak k .
- Baris 2 mendeklarasikan penggunaan arrayList.
- Baris 3 mendeklarasikan variabel untuk menyimpan total kluster.
- Baris 3 menjabarkan perulangan untuk menghitung jumlah data pada kelas j .
- Baris 4-11 merupakan perulangan untuk menghitung jumlah nilai data pada variabel j .
- Baris 14-17 merupakan perulangan untuk menghitung nilai rata-rata sebagai pusat kluster.

5. Menghitung jarak data ke *centroid* dan pengklasteran

Pada tahap kelima langkah yang dilakukan adalah menghitung jarak data ke *centroid* dan pengklasteran. Algoritme 6 menjabarkan implementasi algoritme menghitung jarak data ke *centroid*.

Algoritme 6 : Menghitung Jarak data ke Centroid	
1	for(int i=0; i<distanceToCentroid.length; i++){
2	for(int j=0; j<distanceToCentroid[i].length;
3	j++){
4	float tempTotalSum=0;
5	for(int l=0; l<data.get(i).size(); l++){
6	tempTotalSum +=
7	Math.pow(data.get(i).get(l) -
8	clusterCentroid[j][l], 2);
9	}
10	distanceToCentroid[i][j]= (float)
11	Math.sqrt(tempTotalSum);
12	}
13	nearestClass = getMinimumClassDistance(i,
14	kelas[i][0]);
15	fnew += distanceToCentroid[i][nearestClass];
16	kelas[i][1] = nearestClass;
17	}

Penjelasan Algoritme 6:

- Baris 1 adalah perulangan untuk menghitung jarak data ke pusat klaster untuk variabel *i*.
- Baris 2-12 merupakan perulangan untuk menghitung jarak data ke pusat klaster dengan menggunakan rumus *Euclidean distance*.
- Baris 13-14 berfungsi untuk menghitung jarak data ke pusat klaster terdekat.
- Baris 15 berfungsi untuk menghitung *f* baru.
- Baris 16 berfungsi untuk menentukan kelas data berdasarkan pusat klaster terdekat.

Algoritme 6 menjabarkan kode sumber untuk menghitung jarak data ke *centroid* dan pengklasteran data berdasarkan jarak data ke pusat klaster terdekat, sehingga pada tahap ini data suatu klaster dapat berpindah ke klaster yang lain.

6. Percabangan

Pada tahap ke-tujuh, langkah yang dilakukan adalah memasuki proses percabangan. Percabangan pada tahap ini menentukan apakah iterasi akan berlanjut atau tidak. Algoritme 7 menjabarkan implementasi percabangan program *K-Means*.

Algoritme 7 : Percabangan K-Means	
1	deltaF = Math.abs(fnew - f);
2	f = fnew;
3	iterasi++;
4	if(deltaF == 0 deltaF < treshold){
5	stop = true;
6	}

Penjelasan Algoritme 7:

- Baris 1 merupakan perhitungan untuk mendapatkan nilai *delta*.
- Baris 2 berfungsi untuk mengisi nilai *f* dengan nilai *fnew*.
- Baris 3 berfungsi untuk perulangan iterasi.
- Baris 4-6 merupakan kondisi jika $\Delta f = 0$ atau $\Delta f < \text{threshold}$ maka iterasi berhenti.

4.2.2 Implementasi algoritme *Isodata*

Dalam implementasi algoritme *Isodata* terdapat beberapa proses yang akan dijalankan oleh sistem. Berikut merupakan bagian-bagian implementasi dari algoritme *Isodata* pada studi kasus Analisis Perbandingan klasterisasi data Kejadian titik api di Wilayah Sumatera pada tahun 2001 hingga 2014.

1. *Input variabel*

Mendeklarasikan nilai K_0 , N_{min} , σ^2_{max} , d_{min} dan *input data*. Algoritme 8 menjabarkan implementasi penentuan variabel nilai K_0 , n_{min} , σ^2_{max} , d_{min} dan muat data.

Algoritme 8 : Input variabel Isodata	
1	private int k0;
2	private int nMin;
3	private int maxIteration;
4	private int maxVar;
5	private int dMin;
6	private void muatData(String lokasi){
7	File csvFile = new File(lokasi);
8	
9	if(csvFile.exists()){
10	try{
11	FileReader fr = new FileReader(csvFile);
12	BufferedReader br = new
13	BufferedReader(fr);
14	String st;
15	int baris = 0;
16	a.clear();
17	while((st = br.readLine()) != null){
18	if(baris>0){
19	String[] data = st.split(",");
20	float latitude =
21	Float.parseFloat(data[1]);
22	float longitude =
23	Float.parseFloat(data[2]);
24	float brigness =
25	Float.parseFloat(data[3]);
26	

Algoritme 8 : Input Variabel Isodata (lanjutan)	
27	Date date = Date.valueOf(data[6]);
28	int convidence =
29	Integer.parseInt(data[9]);
30	a.add(new klasterIsodata(latitude,
31	longitude, brigness, date, convidence));
32	}
33	baris++;
34	}
35	String header[] = {"Latitude", "Longitude",
36	"Brigness", "Acq_Date", "Convidence"};
37	DefaultTableModel model = new
38	DefaultTableModel(null, header);
39	model.setColumnCount(5);
40	model.setRowCount(baris-1);
41	for (int i=0; i<(baris-1); i++){
42	model.setValueAt(a.get(i).latitude, i,
43	0);
44	model.setValueAt(a.get(i).longitude,
45	i, 1);
46	model.setValueAt(a.get(i).brigness, i,
47	2);
48	model.setValueAt(a.get(i).acq_date, i,
49	3);
50	model.setValueAt(a.get(i).convidence,
51	i, 4);
52	}
53	jTableRepresentasiData.setModel(model);
54	
55	}catch (Exception e){
56	JOptionPane.showMessageDialog(null, "Error
57	"+e);
58	}
59	}else{
60	JOptionPane.showMessageDialog(null, "File does
61	not exist");
62	}
63	}

Penjelasan Algoritme 8:

- Baris 1-5 adalah deklarasi variabel k_0 , n_{min} , iterasi max, σ^2_{max} , dan d_{min} pada tipe data integer.
- Baris 6 merupakan kode fungsi untuk memuat data.
- Baris 7 berfungsi untuk menjalankan objek untuk file CSV.
- Baris 9-60 adalah kondisi ketika file csv tersedia sehingga program dapat membaca data dengan format csv. Pada proses ini data difilter berdasarkan fitur yang dibutuhkan. Pada tahapan ini fitur yang dipilih adalah latitude, longitude, brightness, acq_date, convidence.
- Baris 61-63 adalah kondisi lain apabila kondisi pada baris ke 9 tidak terpenuhi.

2. Melakukan normalisasi data

Pada tahap yang kedua langkah yang dilakukan adalah melakukan normalisasi data. Metode yang dipilih dalam melakukan normalisasi adalah *min max* normalisasi sama halnya normalisasi data pada algoritme K-Means yang dijabarkan pada algoritme 2.

3. Mengalokasikan data pada kluster

Pada tahap ketiga proses yang dilakukan adalah mengalokasikan data pada kluster secara acak. Algoritme 9 menjabarkan implementasi algoritme untuk mengalokasi data secara acak.

Algoritme 9 : Acak kelas Isodata	
1	for(int i=0; i<a.size(); i++){
2	kelas[i] =0+(int) (Math.random()*jumlahKelas);
3	}

Penjelasan Algoritme 9:

Baris 1 merupakan perulangan variabel sebanyak data < a.

Baris 2 berfungsi untuk menetapkan kelas secara acak.

4. Menghilangkan kluster

Pada tahap keempat, langkah yang dilakukan adalah menghilangkan kluster yang memiliki kluster kurang dari n_{\min} . Tabel 4.14 menjabarkan implementasi menghilangkan kluster yang kurang dari n_{\min}

Algoritme 10 : menghilangkan kluster	
1	if(totalClusterX.get(i).get(1)<nMin){
2	jTextAreaHasil.append("Remove
3	cluster"+(i)+"\n");
4	totalClusterX.remove(i);
5	for(int j=0;
6	j<distanceToCentroid.size(); j++){
7	distanceToCentroid.get(j).remove(i);
8	}
9	for(int j=0; j<kelas.length;
10	j++){
11	if(kelas[j]==i){
12	kelas[j]=
13	getMinimumClassDistance(j);
14	}
15	}
16	updateTotalClusterX(kelas);
17	i--;
18	}

Penjelasan Algoritme 10:

- Baris 1 menjabarkan kondisi jika data i kurang dari n_{\min}
- Baris 2-4 berfungsi untuk menghapus kluster
- Baris 5-9 adalah perulangan untuk menghapus data ke i

- d. Baris 10-14 adalah perulangan untuk mengalokasikan data pada klaster berdasarkan jarak ke pusat klaster terdekat
 - e. Baris 17 berfungsi untuk memperbarui total klaster
 - f. Baris 18 berfungsi untuk mengurangi jumlah klaster
5. Melakukan pembaruan pusat klaster *Isodata*

Pada tahap kelima langkah yang dilakukan adalah melakukan pembaruan nilai rata-rata. Algoritme 11 menjabarkan tentang implementasi pembaruan data.

Algoritme 11 : Pembaruan data	
1	for(int j=0; j<clusterCentroid.get(i).size(); j++){
2	clusterCentroid.get(i).set(j,
3	clusterCentroid.get(i).get(j)/totalClusterX.get(i).get(1
4));
	}

Penjelasan Algoritme 11:

- a. Baris 1 merupakan perulangan sebanyak pusat klaster
 - b. Baris 2-3 berfungsi untuk menghitung nilai rata-rata sebagai pusat klaster
6. *Split (Isodata)*

Pada pada tahap ke enam langkah yang dilakukan adalah melalui proses *split*. Proses *split* adalah proses untuk melakukan pemisahan klaster yang memenuhi syarat untuk dilakukan *split*. Algoritme 12 menjabarkan implementasi proses *split*.

Tabel 4.1 Implementasi proses *Split Isodata*

Algoritme 12 : Split	
1	if(totalClusterX.size()<(k0/2)){
2	for(int i=0;
3	i<totalClusterX.size();i++){
4	double varianMax=getVarianMax(i,
5	kelas, totalClusterX.get(i).get(1));
6	if(varianMax>maxVar &&
7	totalClusterX.get(i).get(1)>2*nMin){
8	jTextAreaHasil.append("Split
9	cluster "+(i)+"\n");
10	System.out.println("Splite
11	cluster"+(i));
12	totalClusterX.add(i+1, new
13	ArrayList<Integer>());
14	ArrayList<ArrayList<Double>>m=
15	new ArrayList<ArrayList<Double>>());
16	for(int j=0;
17	j<data.get(0).size(); j++){
18	double mean
19	=getMeanOrVarian(i, j,
20	kelas, totalClusterX.get(i).get(1), true);
21	double
22	variance=getMeanOrVarian(i, j, kelas,
23	totalClusterX.get(i).get(1), false);

Algoritme 12 : Split (lanjutan)

```

24 m.add(new ArrayList<Double>());
25 m.get(j).add(mean- (Math.sqrt (variance)));
26 m.get(j).add(mean+ (Math.sqrt (variance)));
27 }
28 for(int j=0; j<data.size();
29 j++){
30 for(int k=0;
31 k<m.get(0).size(); k++){
32 double
33 tempDistanceNewClass=0;
34 for(int l=0;
35 l<data.get(j).size(); l++){
36 if(kelas[j]==i){
37 tempDistanceNewClass+=Math.pow(data.get(j).get(l)-
38 m.get(k).get(l), 2);
39 }
40 }
41 tempDistanceNewClass=Math.sqrt(tempDistanceNewClass);
42 if(i%2==0){
43 distanceToCentroid.get(j).set(i,
44 tempDistanceNewClass);
45 }else{
46 distanceToCentroid.get(j).add(i+1,
47 tempDistanceNewClass);
48 }
49 }
50 i--;
51 }
52 }
53 }
54 }
55 }
56 }
57 }

```

Penjelasan Algoritme 12:

- Baris 1 merupakan kondisi dimana syarat untuk melakukan split terpenuhi
- Baris 2-3 merupakan perulangan untuk melakukan pengecekan kondisi yang menjadi syarat kedua untuk *split*.
- Baris 4-5 berfungsi untuk mendapatkan nilai varian max
- Baris 6-7 merupakan kondisi jika syarat kedua terpenuhi
- Baris 8-11 merupakan keluaran dari split
- Baris 12-13 berfungsi untuk melakukan penambahan kluster
- Baris 14-15 berfungsi untuk menyimpan nilai m+ dan m-
- Baris 16-28 merupakan perulangan untuk menghitung nilai m+ dan m-
- Baris 29-56 merupakan perulangan untuk menghitung jarak setiap data ke m+ dan m- dengan menggunakan rumus *Euclidean distance*.

7. Merge

Pada tahap ke-tujuh, langkah yang dilakukan adalah menjalankan proses *merge*. Proses *merge* adalah proses untuk melakukan penggabungan kluster yang memenuhi syarat untuk dilakukan *merge*. Algoritme menjabarkan implementasi proses *merge*.

Algoritme 12 : Merge	
1	else if(totalClusterX.size()>(k0*2)){
2	double[] distance=new
3	double[(totalClusterX.size()*(totalClusterX.size()-
4	1))/2];
5	int index = 0;
6	for(int i=0; i<totalClusterX.size(); i++){
7	for (int j= i+1; j
8	< totalClusterX.size() - 1; j++){
9	double temp=0;
10	for(int k=0; k<clusterCentroid.get(i).size(); k++){
11	k<clusterCentroid.get(i).size(); k++){
12	temp
13	+= Math.pow(clusterCentroid.get(i).get(k)-
14	clusterCentroid.get(j).get(k), 2);
15	}
16	distance[index]=Math.sqrt(temp);
17	distance[index]=Math.sqrt(temp);
18	}else{
19	distance[index]=-1;
20	}
21	index++;
22	}
23	index=0;
24	for(int i=0; i<totalClusterX.size(); i++){
25	boolean get = false;
26	for(int j=i+1; j<totalClusterX.size();
27	i++){
28	if(distance[i]<dMin){
29	jTextAreaHasil.append("Merge
30	Cluster"+(i)+"&"+(j)+"\n");
31	System.out.println("Merge
32	Cluster"+(i)+"&"+(j));
33	for(int k=0; k<clusterCentroid.get(i).size(); k++){
34	double
35	temp1=1/(totalClusterX.get(i).get(1)+totalClusterX.get(j).
36	get(1));
37	double
38	temp2=(totalClusterX.get(i).get(1)*clusterCentroid.get(i).
39	get(k))
40	+(totalClusterX.get(j).get(1)*clusterCentroid.get(j).get(k
41));
42	clusterCentroid.get(i).set(k, temp1*temp2);
43	}
44	get=true;
45	clusterCentroid.remove(j);
46	totalClusterX.get(i).set(1,
47	totalClusterX.get(i).get(1)+totalClusterX.get(j).get(1));
48	
49	
50	
51	totalClusterX.remove(j);
52	for(int k=0; k<data.size();

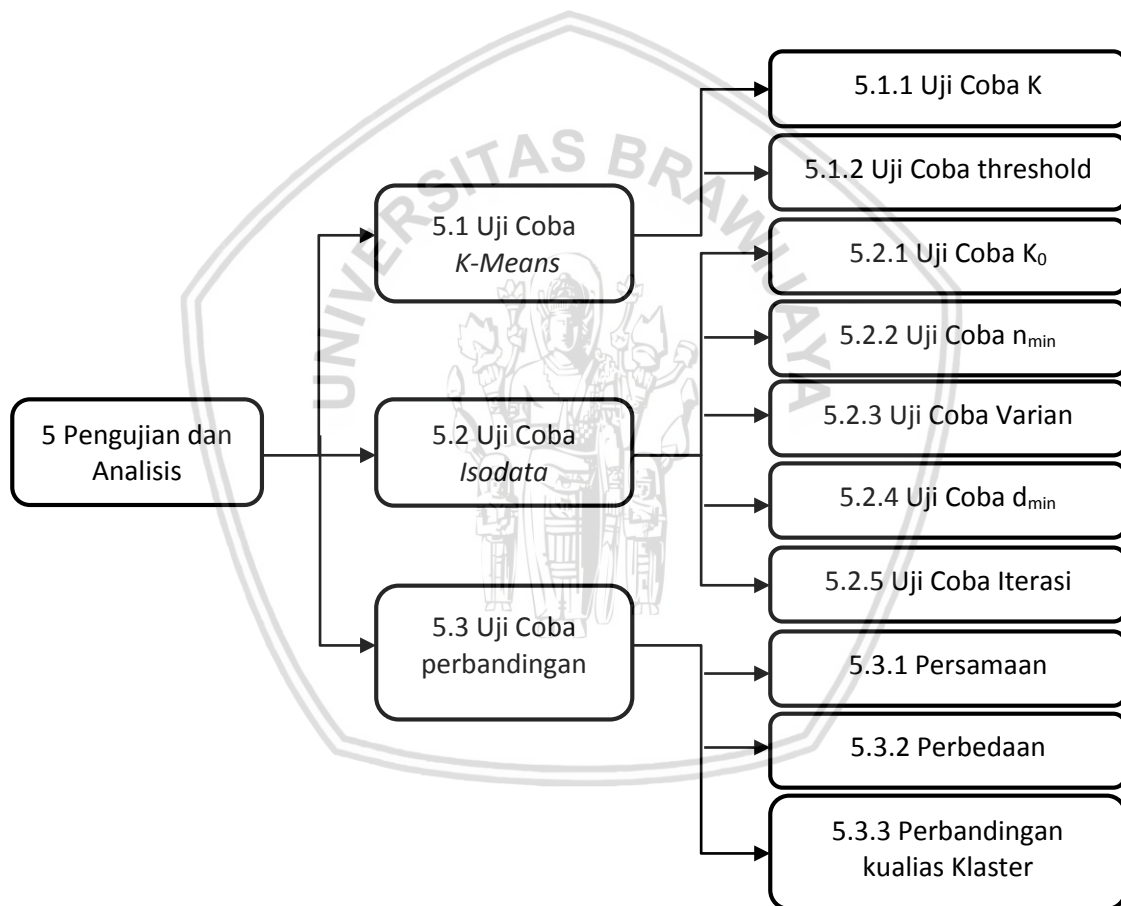
Algoritme 12 : Merge (lanjutan)	
53	k++) {
54	if (kelas[k]==j) {
55	kelas[k]=i;
56	}
57	
58	distanceToCentroid.get(k).remove(j);
59	}
60	}
61	index++;
62	}
63	}
64	}
65	}

Penjelasan Algoritme 12:

- Baris 1 merupakan kondisi dimana syarat untuk melakukan merge terpenuhi
- Baris 2-4 berfungsi untuk menyimpan jarak antar klaster
- Baris 5 merupakan deklarasi variabel index dengan nilai 0
- Baris 6 merupakan perulangan untuk menghitung jarak dari klaster ke i
- Baris 7-8 merupakan perulangan untuk menghitung jarak ke klaster j
- Baris 9 merupakan deklarasi variabel temp dengan nilai 0
- Baris 10-21 berfungsi untuk menghitung jarak antar klaster dengan menggunakan rumus *Euclidean distance*.
- Baris 24 mengalokasikan nilai indeks =0
- Baris 25 merupakan perulangan untuk mengecek kondisi jika terdapat jarak yang kurang dari d_{min} .
- Baris 26 berfungsi untuk menandai jarak yang kurang dari d_{min} .
- Baris 27-62 merupakan perulangan untuk menggabungkan dua klaster yang memenuhi syarat untuk penggabungan klaster dengan cara menghitung pusat klaster baru diantara dua klaster yang memenuhi syarat untuk digabungkan.

BAB 5 PENGUJIAN DAN ANALISIS

Pada bab ini membahas tentang pengujian dan analisis terhadap hasil implementasi yang telah dibahas pada Bab 4. Dalam penelitian ini, pengujian dilakukan dengan melakukan uji coba pada masukan variabel yang ada pada program *K-Means* dan *Isodata*. Masukan variabel yang diuji, diukur berdasarkan hasil nilai *Shilhouette Coefficient*. Dalam hal ini bahasan pengujian yang diangkat adalah pengaruh variabel masukan algoritme *K-Means* dan *Isodata* terhadap nilai SC dan analisis perbandingan kedua metode tersebut. Data yang digunakan pada uji coba, merupakan data sampel sebanyak 10 ribu data dari 289.257 data kejadian titik api. Gambar 5.1 menyajikan cakupan uji coba pada penelitian ini.



Gambar 5.1 Uji Coba

Gambar 5.1 menyajikan gambaran dari cakupan uji coba yang dilakukan dalam sub bab ini. Algoritme *K-Means* dan *Isodata* memiliki variabel masukan yang berbeda, sehingga dalam hal ini pengujian dilakukan dengan melakukan uji coba satu persatu variabel masukan yang ada pada algoritme *K-Means* dan *Isodata*.

5.1 Uji coba *K-Means*

Pada algoritme *K-Means* variabel yang di uji adalah K dan *threshold*. Variabel K dan *threshold* diuji dengan beberapa nilai yang memungkinkan untuk dijadikan sebagai nilai masukan variabel. Dari beberapa nilai masukan akan dipilih salah satu untuk dijadikan variabel masukan pada uji coba pada tahap setelahnya. Nilai yang terpilih adalah nilai yang memiliki nilai rata-rata SC tertinggi.

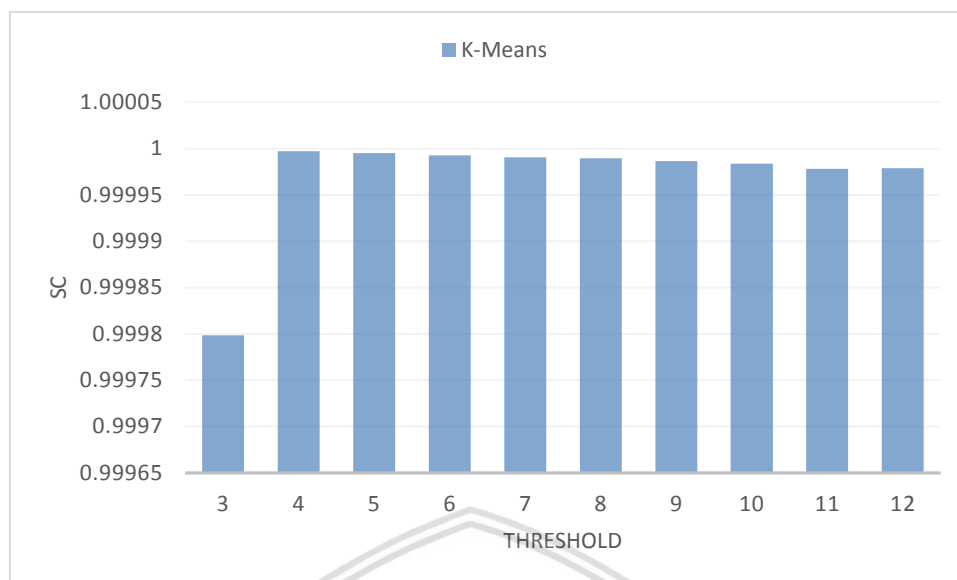
5.1.1 Uji coba K

Uji coba k dilakukan dengan cara memasukkan beberapa nilai masukan untuk variabel k dan satu nilai untuk variabel lainnya. Setiap uji coba pada satu nilai K , dilakukan lima kali percobaan dengan nilai variabel yang sama. Tabel 5.1 menyajikan hasil dari uji coba variabel K .

Tabel 5.1 Uji coba variabel K

No	K	p1	p2	p3	p4	p5	rata-rata
1	3	0.9997986	0.9997984	0.9997985	0.9997986	0.9997986	0.9997985
2	4	0.9999969	0.9999974	0.9999975	0.9999971	0.999997	0.9999972
3	5	0.9999953	0.9999953	0.9999948	0.9999951	0.9999954	0.9999952
4	6	0.9999932	0.9999923	0.9999915	0.9999954	0.9999915	0.9999928
5	7	0.9999911	0.9999908	0.9999902	0.9999912	0.9999889	0.9999904
6	8	0.9999923	0.9999891	0.9999885	0.999984	0.9999931	0.9999894
7	9	0.9999866	0.999988	0.9999847	0.9999871	0.9999853	0.9999863
8	10	0.9999827	0.9999861	0.9999806	0.9999818	0.9999874	0.9999837
9	11	0.9999726	0.9999848	0.9999755	0.9999818	0.9999767	0.9999783
10	12	0.9999738	0.9999683	0.9999774	0.9999905	0.999985	0.999979
Max							0.9999972

Pada Tabel 5.1 uji coba nilai K dilakukan dengan memasukkan 10 nilai yang berbeda. Setiap nilai masukan K diuji dengan lima kali percobaan yang disimbolkan dengan huruf p1 untuk percobaan pertama, p2 untuk percobaan kedua dan seterusnya. Pada uji coba variabel K nilai *threshold* yang dimasukkan adalah 0.5. Berdasarkan hasil dari lima kali percobaan, didapatkan nilai rata-rata SC untuk salah satu nilai variabel masukan. Dari 10 nilai rata-rata SC, nilai yang tertinggi adalah 0.9999972 untuk $K=4$. Hasil yang diperoleh dapat dipengaruhi oleh variasi nilai *threshold* dan data yang digunakan. Sehingga, hasil uji coba dapat berubah berdasarkan variasi nilai *threshold* dan data yang digunakan. Gambar 5.2 merupakan grafik yang dihasilkan dari uji coba pada Tabel 5.1.



Gambar 5.2 Grafik pengaruh k terhadap SC

5.1.2 Uji coba *Threshold*

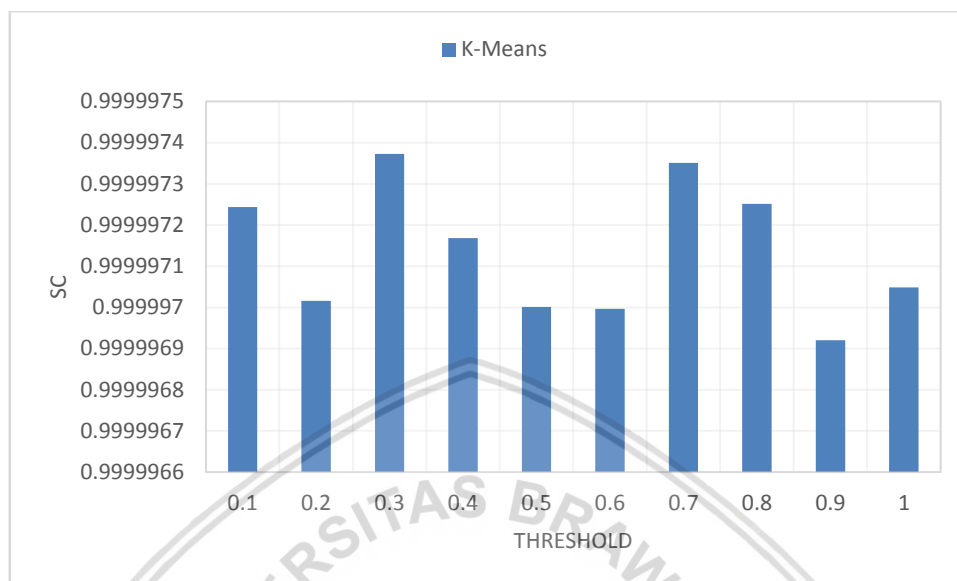
Uji coba *threshold* dilakukan dengan cara memasukkan beberapa nilai masukan untuk variabel *threshold* dan satu nilai untuk variabel lainnya. Setiap uji coba pada satu nilai *threshold*, dilakukan lima kali percobaan dengan nilai variabel yang sama. Tabel 5.2 menyajikan hasil dari uji coba variabel *threshold*.

Tabel 5.2 uji coba variabel *Threshold*

No	Threshold	p1	p2	p3	p4	p5	rata-rata
1	0.1	0.999997	0.9999975	0.9999974	0.9999972	0.9999971	0.999997244
2	0.2	0.999997	0.999997	0.999997	0.9999968	0.9999973	0.999997016
3	0.3	0.999997	0.9999974	0.9999975	0.9999974	0.9999975	0.999997373
4	0.4	0.999997	0.9999971	0.9999974	0.9999969	0.9999974	0.999997169
5	0.5	0.9999969	0.9999972	0.999997	0.9999971	0.9999968	0.999997001
6	0.6	0.9999969	0.9999972	0.999997	0.9999969	0.9999969	0.999996996
7	0.7	0.9999966	0.9999974	0.9999973	0.9999986	0.999997	0.999997351
8	0.8	0.9999972	0.9999975	0.9999971	0.9999974	0.999997	0.999997251
9	0.9	0.9999969	0.999997	0.9999969	0.999997	0.9999968	0.999996921
10	1	0.9999972	0.999997	0.999997	0.9999971	0.9999969	0.999997049
Max							0.999997373

Pada Tabel 5.2 uji coba nilai *threshold* dilakukan dengan memasukkan 10 nilai yang berbeda. Setiap nilai masukan *threshold* diuji dengan lima kali percobaan. Pada uji coba variabel *threshold* nilai *K* yang dimasukkan adalah 4. Berdasarkan hasil dari lima kali percobaan, didapatkan nilai rata-rata SC untuk salah satu nilai variabel masukan. Dari 10 nilai rata-rata SC, nilai yang tertinggi adalah 0.999997373 untuk *threshold* = 0.3. Hasil yang diperoleh dapat dipengaruhi oleh variasi nilai *K* dan data yang digunakan. Sehingga, hasil uji coba dapat berubah

berdasarkan variasi nilai K dan data yang digunakan. Gambar 5.3 merupakan grafik yang dihasilkan dari uji coba pada Tabel 5.2.



Gambar 5.3 Grafik pengaruh *threshold* terhadap SC

5.2 Uji coba *Isodata*

Pada algoritme *Isodata* variabel yang di uji adalah K_0 , n_{min} , σ^2_{max} , d_{min} dan *Iterasi Max*. Variabel tersebut diuji dengan beberapa nilai yang memungkinkan untuk dijadikan sebagai nilai masukan variabel. Dari beberapa nilai masukan akan dipilih salah satu untuk dijadikan variabel masukan pada uji coba pada tahap setelahnya. Nilai yang terpilih adalah nilai yang memiliki nilai rata-rata SC tertinggi.

5.2.1 Uji coba K_0

Uji coba k_0 dilakukan dengan cara memasukkan beberapa nilai masukan untuk variabel k_0 dan satu nilai untuk variabel lainnya. Setiap uji coba pada satu nilai K_0 , dilakukan lima kali percobaan dengan nilai variabel yang sama. Nilai variabel lain yang dimasukkan adalah $n_{min} = 100$, $\sigma^2_{max} = 0.9$, $d_{min} = 0.5$ dan *Iterasi Max* = 50. Tabel 5.3 menyajikan hasil dari uji coba variabel K_0 .

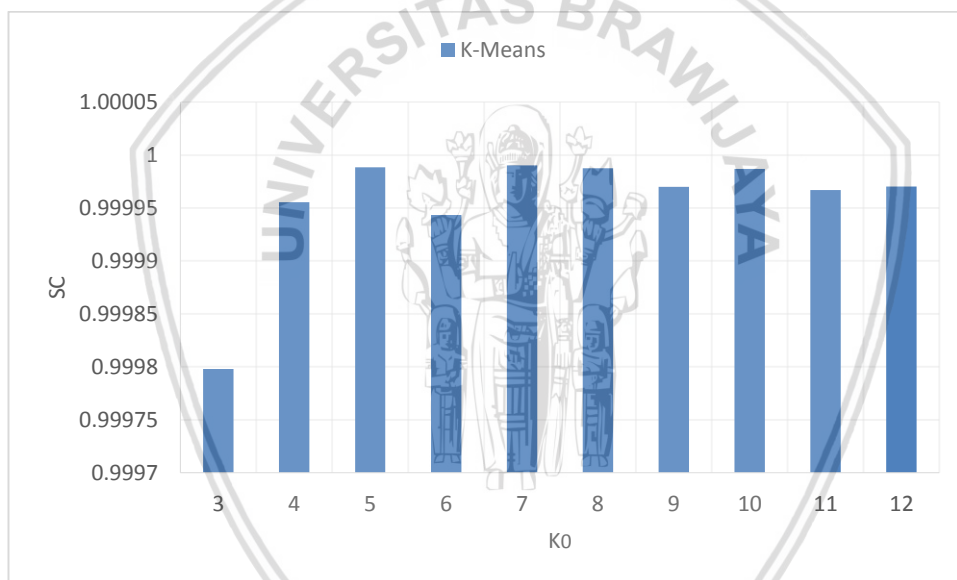
Tabel 5.3 Uji coba K_0

No	K_0	p1	p2	p3	p4	p5	rata- rata
1	3	0.999798	0.999799	0.999797	0.999798	0.999799	0.99979809
2	4	0.999799	0.999995	0.999992	0.999996	0.999996	0.99995546
3	5	0.999993	0.999988	0.999994	0.999984	0.999982	0.99998847
4	6	0.999985	0.999984	0.999971	0.999978	0.999799	0.99994345
5	7	0.999984	0.999988	0.999993	0.99999	0.999995	0.99999024
6	8	0.999998	0.999982	0.999984	0.99998	0.999993	0.99998751
7	9	0.999978	0.999993	0.999975	0.999935	0.999969	0.99997006
8	10	0.999959	0.999995	0.999995	0.999997	0.999988	0.9999869

Tabel 5.3 Uji coba K_0 (lanjutan)

No	K_0	p1	p2	p3	p4	p5	rata- rata
9	11	0.999993	0.999982	0.999931	0.99996	0.999967	0.99996683
10	12	0.99995	0.999971	0.999992	0.999995	0.999944	0.99997025
Max							0.99999024

Pada Tabel 5.3 uji coba nilai K_0 dilakukan dengan memasukkan 10 nilai yang berbeda. Setiap nilai masukan K_0 , diuji dengan lima kali percobaan. Berdasarkan hasil dari lima kali percobaan, didapatkan nilai rata-rata SC untuk salah satu nilai variabel masukan. Dari 10 nilai rata-rata SC, nilai yang tertinggi adalah 0.99999024 untuk $K_0 = 7$. Hasil yang diperoleh dapat dipengaruhi oleh variasi nilai n_{min} , σ^2_{max} , d_{min} , Iterasi Max dan data yang digunakan. Sehingga, hasil uji coba dapat berubah berdasarkan variasi nilai K dan data yang digunakan. Gambar 5.4 merupakan grafik yang dihasilkan dari uji coba K_0 pada Tabel 5.3.



Gambar 5.4 Grafik pengaruh K_0 terhadap SC

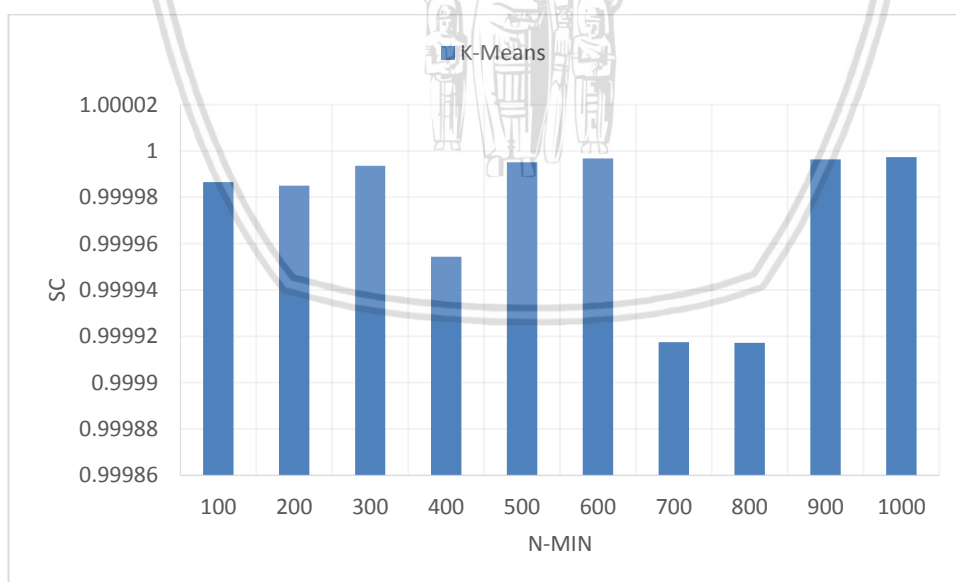
5.2.2 Uji coba n_{min}

Uji coba n_{min} dilakukan dengan cara memasukkan beberapa nilai masukan untuk variabel n_{min} dan satu nilai untuk variabel lainnya. Setiap uji coba pada satu nilai n_{min} , dilakukan lima kali percobaan dengan nilai variabel yang sama. Nilai variabel lain yang dimasukkan adalah $K_0 = 7$, $\sigma^2_{max} = 0.9$, $d_{min} = 0.5$ dan Iterasi Max = 50. Tabel 5.4 menyajikan hasil dari uji coba variabel n_{min} .

Tabel 5.4 Uji coba n_{min}

No	Nmin	p1	p2	p3	p4	p5	rata- rata
1	100	0.999994	0.999983	0.999991	0.999993	0.999972	0.9999866
2	200	0.999994	0.999985	0.999992	0.999957	0.999997	0.999985043
3	300	0.999997	0.999988	0.999993	0.999996	0.999993	0.999993602
4	400	0.999798	0.999989	0.999997	0.999993	0.999995	0.999954407
5	500	0.999995	0.999995	0.999993	0.999996	0.999997	0.999995114
6	600	0.999996	0.999999	0.999995	0.999997	0.999997	0.99999678
7	700	0.999798	0.999996	0.999997	0.999798	0.999997	0.999917432
8	800	0.999995	0.999799	0.999998	0.999995	0.999799	0.999917173
9	900	0.999995	0.999996	0.999998	0.999995	0.999998	0.999996399
10	1000	0.999999	0.999997	0.999998	0.999996	0.999997	0.999997315
Max							0.999997315

Pada Tabel 5.4 uji coba nilai n_{min} , dilakukan dengan memasukkan 10 nilai yang berbeda. Setiap nilai masukan K_0 , diuji dengan lima kali percobaan. Berdasarkan hasil dari lima kali percobaan, didapatkan nilai rata-rata SC untuk salah satu nilai variabel masukan. Dari 10 nilai rata-rata SC, nilai yang tertinggi adalah 0.999997315 untuk $n_{min} = 1000$. Hasil yang diperoleh dapat dipengaruhi oleh variasi nilai K_0 , σ^2_{max} , d_{min} , Iterasi Max dan data yang digunakan. Sehingga, hasil uji coba dapat berubah berdasarkan variasi nilai K dan data yang digunakan. Gambar 5.5 merupakan grafik yang dihasilkan dari uji coba n_{min} pada Tabel 5.4.



Gambar 5.5 Grafik pengaruh n_{min} terhadap SC

5.2.3 Uji coba σ^2_{max}

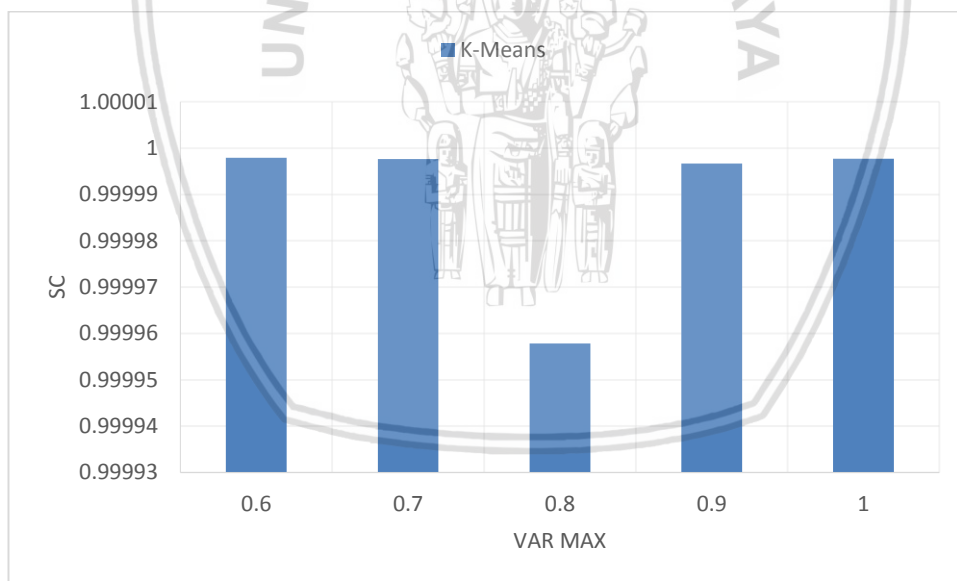
Uji coba variabel σ^2_{max} dilakukan dengan cara memasukkan beberapa nilai masukan untuk variabel σ^2_{max} dan satu nilai untuk variabel lainnya. Setiap uji coba

pada satu nilai σ^2_{max} , dilakukan lima kali percobaan dengan nilai variabel yang sama. Nilai variabel lain yang dimasukkan adalah $K_0 = 7$, $n_{min} = 1000$, $d_{min} = 0.3$ dan $Iterasi\ Max = 50$. Tabel 5.5 menyajikan hasil dari uji coba variabel σ^2_{max} .

Tabel 5.5 Uji coba σ^2_{max}

No	VM	p1	p2	p3	p4	p5	rata- rata
1	0.6	0.999999	0.999996	0.999999	0.999998	0.99999868	0.999997904
2	0.7	0.999995	0.999999	0.999999	0.999997	0.99999799	0.999997613
3	0.8	0.999798	0.999996	0.999999	0.999999	0.99999833	0.999957847
4	0.9	0.999997	0.999995	0.999997	0.999997	0.99999639	0.999996672
5	1	0.999998	0.999999	0.999997	0.999997	0.99999776	0.999997677
Max							0.999997904

Pada Tabel 5.5 uji coba nilai σ^2_{max} , dilakukan dengan memasukkan 5 nilai yang berbeda. Setiap nilai masukan K_0 , diuji dengan lima kali percobaan. Berdasarkan hasil dari lima kali percobaan, didapatkan nilai rata-rata SC untuk salah satu nilai variabel masukan. Dari 5 nilai rata-rata SC, nilai yang tertinggi adalah 0.999997904 untuk $\sigma^2_{max} = 0.6$. Hasil yang diperoleh dapat dipengaruhi oleh variasi nilai K_0 , n_{min} , d_{min} , $Iterasi\ Max$ dan data yang digunakan. Gambar 5.6 merupakan grafik yang dihasilkan dari uji coba n_{min} pada Tabel 5.5.



Gambar 5.6 Grafik pengaruh σ^2_{max} terhadap SC

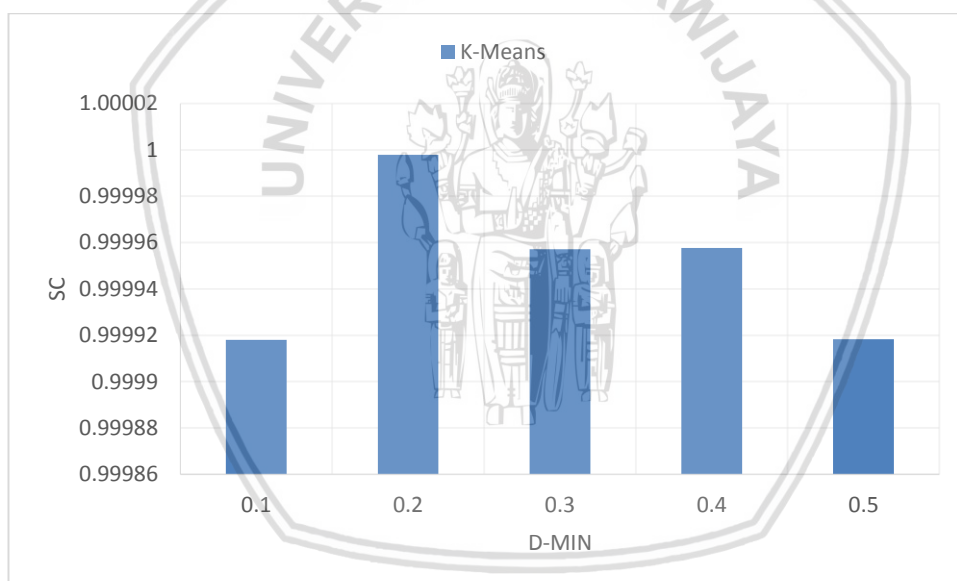
5.2.4 Uji coba d_{min}

Uji coba variabel d_{min} dilakukan dengan cara memasukkan beberapa nilai masukan untuk variabel d_{min} dan satu nilai untuk variabel lainnya. Setiap uji coba pada satu nilai variabel d_{min} , dilakukan lima kali percobaan dengan nilai variabel yang sama. Nilai variabel lain yang dimasukkan adalah $K_0 = 7$, $n_{min} = 1000$, $\sigma^2_{max} = 0.3$ dan $Iterasi\ Max = 50$. Tabel 5.6 menyajikan hasil dari uji coba variabel d_{min} .

Tabel 5.6 Uji coba d_{min}

No	d_{min}	p1	p2	p3	p4	p5	rata- rata
1	0.1	0.999999	0.999798	0.999799	0.999997	0.99999767	0.99991799
2	0.2	0.999997	0.999998	0.999999	0.999998	0.99999842	0.99999791
3	0.3	0.999798	0.999996	0.999997	0.999997	0.99999675	0.99995713
4	0.4	0.999998	0.999998	0.999799	0.999997	0.99999676	0.99995766
5	0.5	0.999997	0.999798	0.999998	0.999999	0.9997995	0.99991824
Max							0.99999791

Pada Tabel 5.6 uji coba nilai d_{min} , dilakukan dengan memasukkan 5 nilai yang berbeda. Setiap nilai masukan d_{min} , diuji dengan lima kali percobaan. Berdasarkan hasil dari lima kali percobaan, didapatkan nilai rata-rata SC untuk salah satu nilai variabel masukan. Dari 5 nilai rata-rata SC, nilai yang tertinggi adalah 0.99999791, untuk $d_{min} = 0.2$. Hasil yang diperoleh dapat dipengaruhi oleh variasi nilai K_0 , n_{min} , σ^2_{max} , *Iterasi Max* dan data yang digunakan. Gambar 5.7 merupakan grafik yang dihasilkan dari uji coba d_{min} pada Tabel 5.6.



Gambar 5.7 Grafik pengaruh d_{min} terhadap SC

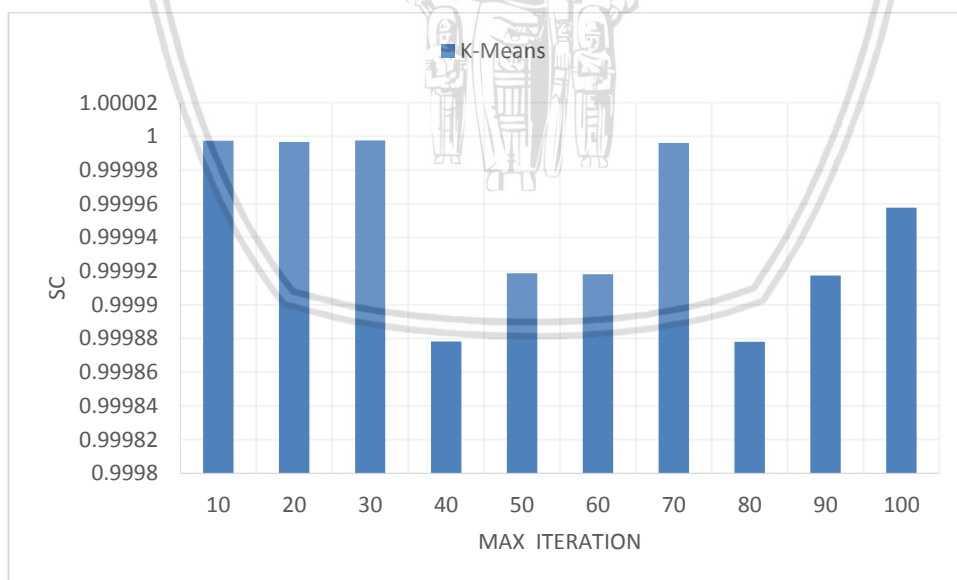
5.2.5 Uji coba *Iterasi Max*

Uji coba variabel iterasi max dilakukan dengan cara memasukkan beberapa nilai masukan untuk variabel *iterasi max* dan satu nilai untuk variabel lainnya. Setiap uji coba pada satu nilai variabel *iterasi max*, dilakukan lima kali percobaan dengan nilai variabel yang sama. Nilai variabel lain yang dimasukkan adalah $K_0 = 7$, $n_{min} = 1000$, $\sigma^2_{max} = 0.6$ dan $d_{min} = 0.2$. Tabel 5.7 menyajikan hasil dari uji coba variabel *Iterasi Max*.

Tabel 5.7 Uji coba Iterasi Max

No	IM	p1	p2	p3	p4	p5	rata- rata
1	10	0.999999	0.999996	0.999997	0.999996	0.99999872	0.99999745
2	20	0.999996	0.999996	0.999996	0.999997	0.99999851	0.9999967
3	30	0.999997	0.999998	0.999998	0.999997	0.99999699	0.9999976
4	40	0.999799	0.999997	0.999799	0.999997	0.99979882	0.99987819
5	50	0.999999	0.999999	0.999799	0.999799	0.99999856	0.99991882
6	60	0.999998	0.999799	0.999799	0.999999	0.99999692	0.99991823
7	70	0.999997	0.999995	0.999996	0.999995	0.99999672	0.99999607
8	80	0.999798	0.999997	0.999799	0.999799	0.9999975	0.99987812
9	90	0.999996	0.999798	0.999997	0.999799	0.99999669	0.99991748
10	100	0.999998	0.999998	0.999997	0.999799	0.99999731	0.99995766
Max							0.9999976

Pada Tabel 5.7 uji coba nilai *Iterasi Max*, dilakukan dengan memasukkan 10 nilai yang berbeda. Setiap nilai masukan *Iterasi Max*, diuji dengan lima kali percobaan. Berdasarkan hasil dari lima kali percobaan, didapatkan nilai rata-rata SC untuk salah satu nilai variabel masukan. Dari 10 nilai rata-rata SC, nilai yang tertinggi adalah 0.9999976, untuk *Iterasi Max* = 30. Hasil yang diperoleh dapat dipengaruhi oleh variasi nilai K_0 , n_{min} , σ^2_{max} , d_{min} dan data yang digunakan. Gambar 5.8 merupakan grafik yang dihasilkan dari uji coba *Iterasi Max* pada Tabel 5.7.



Gambar 5.8 Grafik pengaruh *Iterasi max* terhadap SC

5.3 Analisis perbandingan

Analisis perbandingan merupakan tahapan yang membahas perbandingan kedua metode yang digunakan dalam penelitian ini. Kedua metode yang

dibandingkan adalah *K-Means* dan *Isodata*. Diantara metode *K-Means* dan *Isodata* memiliki persamaan dan perbedaan. Dalam bahasan analisis perbandingan, membahas tentang persamaan, perbedaan dan perbandingan kualitas kluster antara metode *K-Means* dan *Isodata* yang digunakan dalam penelitian ini.

5.3.1 Persamaan

Secara umum metode *K-Means* dan *Isodata* memiliki siklus pengklasteran yang sama. Berikut merupakan penjabaran persamaan antara metode *K-Means* dan *Isodata* yang digunakan dalam penelitian ini.

1. Persamaan pada tahap pengacakan kluster pada langkah awal
2. Persamaan dalam pentuan pusat kluster
3. Persamaan metode dalam menghitung jarak data ke pusat kluster
4. Persamaan dalam mengalokasikan data pada jarak ke pusat kluster terdekat

5.3.2 Perbedaan

Perbedaan yang ada pada metode *K-Means* dan *Isodata* merupakan perbedaan yang menjadi ciri khas masing-masing metode. Berikut merupakan perbedaan antara metode *K-Mean* dan *Isodata*.

1. Pada metode *K-Means* terdapat variabel K , $threshold$, F dan $Delta$ dalam melukaukan klusterisasi, sedangkan pada metode *Isodata* terdapat variabel K_0 , n_{min} , σ^2_{max} , d_{min} dan $Iterasi Max$.
2. Pada metode *K-Means* tidak terdapat penambahan dan pengurangan kluster pada saat proses pengklasteran dijalankan, sedangkan pada metode *Isodata*, memungkinkan terjadinya penambahan atau pengurangan kluster dengan ketentuan yang berlaku.

5.3.3 Perbandingan kualitas kluster

Perbandingan kualitas kluster merupakan tahapan yang membahas tentang perbandingan metode *K-Means* dan *Isodata* berdasarkan nilai dari kualitas kluster yang dihasilkan dengan metode *Silhouette Coefficient* (SC). Perbandingan kualitas kluster dilakukan dengan cara memasukkan nilai variabel terbaik dengan pengujian 10.000 data yang telah dilakukan sebelumnya. Seluruh variabel terbaik pada metode *K-Means* dan *Isodata* dimasukkan pada masing-masing program dan dilakukan uji coba sebanyak lima kali percobaan. Nilai akhir yang dijadikan sebagai tolak ukur kualitas kluster kedua metode, merupakan nilai rata-rata SC dari lima kali percobaan yang dilakukan. Pada algoritme *K-Means* nilai variabel yang dimasukkan adalah $K=4$ dan $threshold=0.3$. Pada algoritme *Isodata* nilai variabel yang dimasukkan adalah $K_0=7$, $n_{min}=1000$, $\sigma^2_{max}=0.6$, $d_{min}=0.2$ dan $Iterasi Max=30$. Tabel 5.8 menyajikan hasil dari uji coba perbandingan kualitas kluster

Tabel 5.8 perbandingan nilai SC

Metode	p1	p2	p3	p4	p5	rata-rata SC
K-Means	0.999997	0.999997	0.999997	0.999997	0.999997	0.999997187
Isodata	0.999995	0.999998	0.999998	0.999997	0.999799	0.999957161

Pada Tabel 5.8 uji merupakan hasil uji coba yang dilakukan sebanyak lima kali percobaan dengan memasukkan nilai variabel yang sama. Berdasarkan hasil nilai rata-rata yang didapat pada Tabel 5.8, Metode *K-Means* menghasilkan nilai kluster yang lebih tinggi dibandingkan dengan *Isodata* dengan nilai 0.999997187. Hasil yang diperoleh dapat dipengaruhi oleh variasi nilai K dan *threshold* pada Algoritme *K-Means*. Sedangkan pada algoritme *Isodata* hasil yang diperoleh dapat dipengaruhi oleh variasi nilai K_0 , n_{min} , σ^2_{max} , d_{min} dan *Iterasi Max*. Apabila diamati dari segi jumlah variabel algoritme *Isodata* memiliki jumlah variabel yang lebih banyak dari pada algoritme *K-Means*. Dari sudut pandang yang lain, apabila diamati dari segi tahapan algoritme, algoritme *Isodata* memiliki tahapan yang lebih kompleks dari pada algoritme *K-Means*. Dalam hal ini algoritme *K-Means* merupakan algoritme yang lebih sederhana dibandingkan algoritme *Isodata*. Kompleksitas yang tidak terlalu dibutuhkan pada algoritme *Isodata* dapat menjadi faktor yang dapat mempengaruhi kualitas kluster algoritme *K-Means* lebih unggul dibandingkan algoritme *Isodata* pada uji coba klusterisasi data kejadian titik api untuk 1000 data.



BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan studi pustaka, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis penelitian yang dilakukan dalam penelitian ini maka dapat ditarik beberapa kesimpulan, yaitu:

1. Algoritme *K-Means* menghasilkan kualitas kluster yang sangat baik berdasarkan nilai SC yang diperoleh. Sehingga metode *K-Means* sangat layak untuk digunakan dalam mengkluster data kejadian titik api di wilayah Sumatera pada tahun 2001 hingga 2014.
2. Algoritme *Isodata* menghasilkan kualitas kluster yang sangat baik berdasarkan nilai SC yang diperoleh. Sehingga metode *Isodata* sangat layak untuk digunakan dalam mengkluster data kejadian titik api di wilayah Sumatera pada tahun 2001 hingga 2014.
3. Berdasarkan hasil pengujian dan analisis yang dilakukan dalam penelitian ini, algoritme *K-Means* menghasilkan nilai kualitas kluster yang lebih tinggi dibandingkan dengan algoritme *Isodata* untuk pengujian 10.000 data. Pada algoritme *K-Means* nilai SC yang diperoleh sebesar 0.999997187 dan algoritme *Isodata* sebesar 0.999957161. Sehingga dalam hal ini, antara algoritme *K-Means* dan *Isodata* memiliki selisih nilai SC yang kecil.

6.2 Saran

Berdasarkan hasil akhir yang didapat dalam penelitian ini penulis, memberikan saran untuk pengembangan selanjutnya, yaitu Pada tahapan pengujian dapat dilakukan dengan skenario pengujian yang berbeda atau jumlah data yang berbeda, sehingga dapat meningkatkan kompleksitas pengujian yang dilakukan.

DAFTAR PUSTAKA

- Agustin, F. E. M., Fitria, A. & Hanifah, A., 2015. Implementasi Algoritma K-Means untuk Menentukan Kelompok. *Jurnal Teknik Informatika*, VIII(1), pp. 73-78.
- Endrawati, 2016. *Analisa Data Titik Panas (Hotspot) dan Area Kebakaran Lahan*. 1 ed. Jakarta: Direktorat Inventarisasi dan Pemantauan Sumber Daya Hutan, Ditjen Planologi Kehutanan dan Tata Lingkungan Kementerian Lingkungan Hidup dan Kehutanan.
- EOSDIS, n.d. *About EOSDIS*. [Online] Available at: <https://earthdata.nasa.gov/about> [Accessed 23 7 2016].
- Ghosh, S. & Dubey, S. K., 2013. Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. *International Journal of Advanced Computer Science and Applications*, IV(4), pp. 35-39.
- Giglio, L., Descloitres, J., Justice, C. O. & Kaufman, Y. J., 2003. An Enhanced Contextual Fire Detection Algorithm for MODIS. *EISEVIER*, pp. 273-282.
- Harrison, M. E., Page, S. E. & Limin, S. H., 2009. The Global Impact of Indonesian Forest. *Journal of Biological Education*, LVI(3), pp. 156-163.
- Liu, Q., Zhao, Z., Li, Y.-X. & Li, Y., 2012. Feature selection based on sensitivity analysis of fuzzy ISODATA. *Neurocomputing*, Issue 87, pp. 29-37.
- Microsoft, n.d. *Datevalue (Fungsi Datevalue)*. [Online] Available at: <https://support.office.com/id-id/article/datevalue-fungsi-datevalue-df8b07d4-7761-4a93-bc33-b7471bbff252> [Accessed 9 April 2018].
- Muningsih, E. & Kiswati, S., 2015. Penerapan Metode K-Means Untuk Clustering Produk Online Shop. *Bianglala Informatika*, III(1), pp. 10-17.
- Parwati, et al., 2016. *Informasi Titik Panas (Hotspot) Kebakaran Hutan/Lahan*, s.l.: Pusat Pemanfaatan Penginderaan Jauh.
- Pramesti, D. F., Furqon, M. T. & Dewi, C., 2017. Implementasi Metode K-Medoids Clustering untuk Pengelompokan Data Potensi Kebakaran Hutan/Lahan Berdasarkan Persebaran Titik Panas. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume I, pp. 723-732.
- Rasyid, F., 2014. Permasalahan dan Dampak Kebakaran Hutan. *Jurnal Lingkar Widyaishwara*, Issue 1, pp. 47-59.
- Rohmawati, N., Defiyanti, S. & Jajuli, M., 2015. Implementasi Algoritma K-Means dalam Pengklasteran Mahasiswa Pelamar Beasiswa. *Jurnal Ilmiah teknologi Informasi Terapan*, I(2), pp. 62-68.
- Santoso, B., Cholissodin, I. & Setiawan, B. D., 2017. Optimasi K-Means untuk Clustering Kinerja Akademik Dosen Menggunakan Algoritme Genetika.

Pengembangan Teknologi Informasi dan Ilmu Komputer, 1(12), pp. 1652-1659.

Sari, M., 2016. *Ilmu Bumi : Garis Bujur*. [Online] Available at: <https://ilmugeografi.com/ilmu-bumi/garis-bujur> [Accessed 25 Oktober 2017].

Sari, M., 2016. *Ilmu Bumi : Garis Lintang*. [Online] Available at: <https://ilmugeografi.com/ilmu-bumi/garis-lintang> [Accessed 25 Oktober 2017].

Wahyuni, I., Yudha, A. A., Asyrofa, R. & Wayan, F. M., 2016. Clustering Nasabah Bank Berdasarkan Tingkat Likuiditas Menggunakan Hybrid. *Jurnal Ilmiah Teknologi dan Informasi ASIA (JITIKA)*, X(2), pp. 24-33.

Wang, R., 2016. *Classification : Unsupervised Clasification-Clustering*. [Online] Available at: <http://fourier.eng.hmc.edu/e161/lectures/classification/node12.html> [Accessed 6 Juni 2016].

Yu, J., Zhou, Z., Zhong, H. & Huang, H., 2012. An Improved Fuzzy ISODATA Algorithm for Credit Risk Assessment of the EIT Enterprises. *Modern Economy*, III(1), pp. 686-689.

Zahrotun, L., 2015. Analisis Pengelompokan Jumlah Penumpang Bus Trans Jogja Menggunakan Metode Clustering K-Means dan Algglomerative Hierarchical Clustering(AHC). *Jurnal Informatika*, IX(1), pp. 1039-1047.

LAMPIRAN A MANUALISASI SILHOUETTE COEFFICIENT

Manualisasi *Silhouette Coefficient* dilakukan dengan 10 data kejadian titik api. Berikut merupakan proses yang dilakukan dalam metode *Silhouette Coefficient*.

1. Menghitung rata-rata jarak antar objek yang berada dalam satu kluster yang sama, dengan persamaan.

$$a(i) = \frac{1}{|A|} \sum_{j \in C} d(i, j)$$

Tabel 1 berikut menyajikan hasil rata-rata jarak antar objek yang berada dalam dalam satu kluster yang sama

Tabel 1 Jarak Antar objek dalam satu kluster

No	F1	F2	F3	F4	F5	Kluster	Rata Rata Jarak
1	1	0	0	0	0	1	0.354949016
2	0.99765945	0.074120083	0.030769231	0.111111111	0.266666667	1	0.233941374
3	0.94051102	0.149068323	0.112820513	0.222222222	0.283333333	1	0.289052433
4	0.395943047	0.93747412	0.315384615	0.333333333	0.7	2	0.816165546
5	0.166569144	1	0.146153846	0.444444444	0.083333333	3	0.523857589
6	0.16110786	0.991304348	1	0.555555556	1	2	0.816165546
7	0.160912815	0.995859213	0.253846154	0.666666667	0.5	3	0.40166852
8	0.000195046	0.967287785	0.033333333	0.777777778	0.333333333	3	0.309019906
9	0	0.963146998	0.035897436	0.888888889	0.333333333	3	0.326469271
10	0.161497952	0.993374741	0.194871795	1	0.3	3	0.392323711

2. Menghitung jarak antar objek yang berbeda kluster dan mengambil nilai yang terkecil, dengan persamaan

$$d(i, C) = \frac{1}{|A|} \sum_{j \in C} d(i, j)$$

Tabel 2 berikut menyajikan hasil rata-rata antar objek yang berbeda kluster dan nilai jarak minimum.

Tabel 2 Jarak antar objek lintas kluster

No	F1	F2	F3	F4	F5	Kalster	Rata2 jarak ke C1	Rata2 jarak ke C2	Rata2 jarak ke C3	Min
1	1	0	0	0	0	1		1.696577672	1.588425413	1.588425413
2	0.99765945	0.074120083	0.030769231	0.111111111	0.266666667	1		1.493621752	1.457773223	1.457773223
3	0.94051102	0.149068323	0.112820513	0.222222222	0.283333333	1		1.360982882	1.324313842	1.324313842
4	0.395943047	0.93747412	0.315384615	0.333333333	0.7	2	1.219494667		0.710859537	0.710859537
5	0.166569144	1	0.146153846	0.444444444	0.083333333	3	1.293624786	0.97445435		0.97445435
6	0.16110786	0.991304348	1	0.555555556	1	2	1.81462687		1.150833906	1.150833906
7	0.160912815	0.995859213	0.253846154	0.666666667	0.5	3	1.409434138	0.683579294		0.683579294
8	0.000195046	0.967287785	0.033333333	0.777777778	0.333333333	3	1.497903229	0.98023486		0.98023486
9	0	0.963146998	0.035897436	0.888888889	0.333333333	3	1.548063234	1.026670811		1.026670811
10	0.161497952	0.993374741	0.194871795	1	0.3	3	1.535162076	0.989294292		0.989294292

3. Menghitung nilai *Silhouette Coefficient* dengan menggunakan persamaan

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Tabel 3 berikut menyajikan nilai SC yang dihasilkan untuk masing-masing data

Tabel 3 Nilai SC

No	F1	F2	F3	F4	F5	Klaster	3. Nilai SC
1	1	0	0	0	0	1	0.776540331
2	0.99765945	0.074120083	0.030769231	0.111111111	0.266666667	1	0.839521422
3	0.94051102	0.149068323	0.112820513	0.222222222	0.283333333	1	0.78173419
4	0.395943047	0.93747412	0.315384615	0.333333333	0.7	2	-0.129025305
5	0.166569144	1	0.146153846	0.444444444	0.083333333	3	0.462409307
6	0.16110786	0.991304348	1	0.555555556	1	2	0.290805092
7	0.160912815	0.995859213	0.253846154	0.666666667	0.5	3	0.41240391
8	0.000195046	0.967287785	0.033333333	0.777777778	0.333333333	3	0.684749116
9	0	0.963146998	0.035897436	0.888888889	0.333333333	3	0.682011734
10	0.161497952	0.993374741	0.194871795	1	0.3	3	0.603430734

Rata- rata nilai SC dari 10 data adalah 0.540458053.

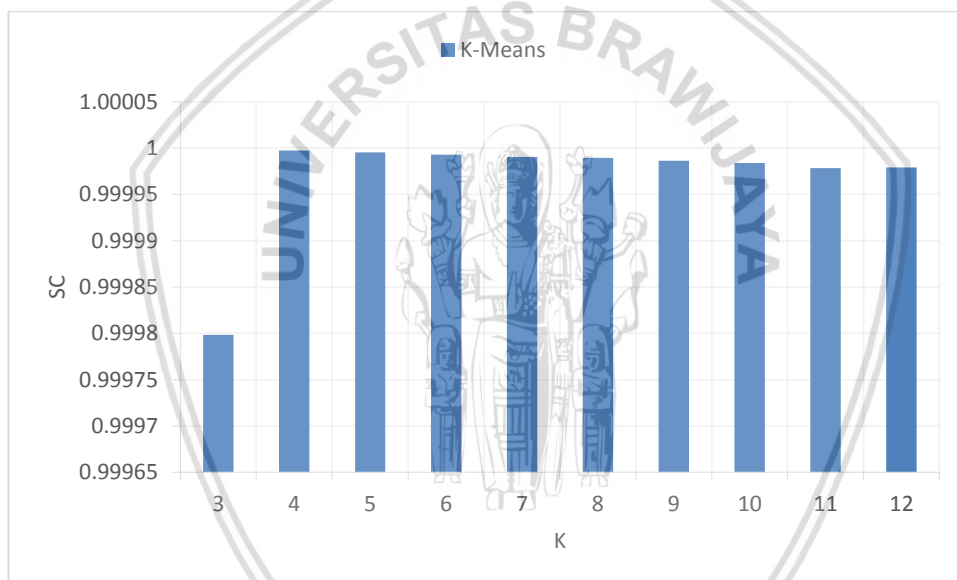


LAMPIRAN B UJI COBA K-MEANS

B.1 Uji coba variabel *K*

No	K	Threshold	Data	p1	p2	p3	p4	p5	rata-rata
1	3	0.5	100000	0.999798565	0.999798386	0.99979854	0.999798584	0.99979858	0.999798531
2	4			0.999996888	0.999997414	0.99999748	0.999997111	0.99999704	0.999997187
3	5			0.999995267	0.999995307	0.9999948	0.999995067	0.9999954	0.999995168
4	6			0.999993215	0.99999229	0.9999915	0.999995398	0.99999154	0.999992787
5	7			0.999991067	0.999990829	0.9999902	0.999991197	0.99998895	0.999990448
6	8			0.999992339	0.999989089	0.99998852	0.999984005	0.99999311	0.999989412
7	9			0.999986593	0.999987957	0.99998466	0.999987134	0.99998532	0.999986332
8	10			0.999982697	0.999986105	0.99998063	0.999981752	0.99998743	0.999983722
9	11			0.999972592	0.999984754	0.99997553	0.999981826	0.99997674	0.999978289
10	12			0.999973844	0.99996831	0.99997736	0.999990523	0.99998504	0.999979015
				Max					0.999997187

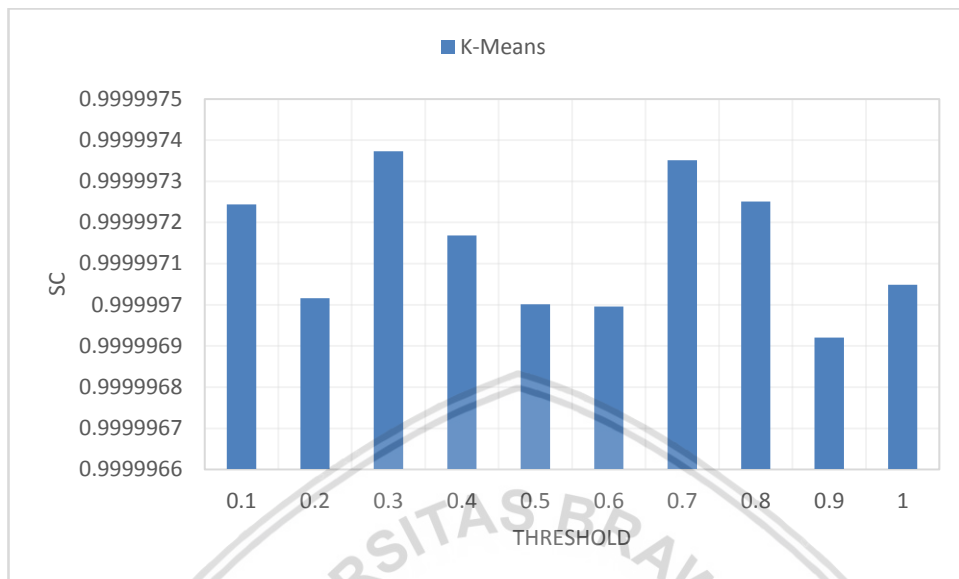
B.2 Grafik uji coba variabel *K*



B.3 Uji coba variabel *Threshold*

No	K	Threshold	Data	p1	p2	p3	p4	p5	rata rata
1	4	0.1	10000	0.999997026	0.99999745	0.99999744	0.999997223	0.99999708	0.999997244
2		0.2		0.999996969	0.999997014	0.99999697	0.999996849	0.99999728	0.999997016
3		0.3		0.99999705	0.999997438	0.99999745	0.99999744	0.99999748	0.999997373
4		0.4		0.999996977	0.999997067	0.99999742	0.999996939	0.99999744	0.999997169
5		0.5		0.999996858	0.999997231	0.99999701	0.999997065	0.99999685	0.999997001
6		0.6		0.999996913	0.999997194	0.99999703	0.999996936	0.99999691	0.999996996
7		0.7		0.999996557	0.999997414	0.99999728	0.999998551	0.99999695	0.999997351
8		0.8		0.999997198	0.999997456	0.99999713	0.999997422	0.99999705	0.999997251
9		0.9		0.999996943	0.999996988	0.99999686	0.999997036	0.99999678	0.999996921
10		1		0.999997183	0.999997014	0.99999704	0.999997119	0.99999689	0.999997049
				Max					0.999997373

B.4 Grafik Uji coba Variabel *threshold*

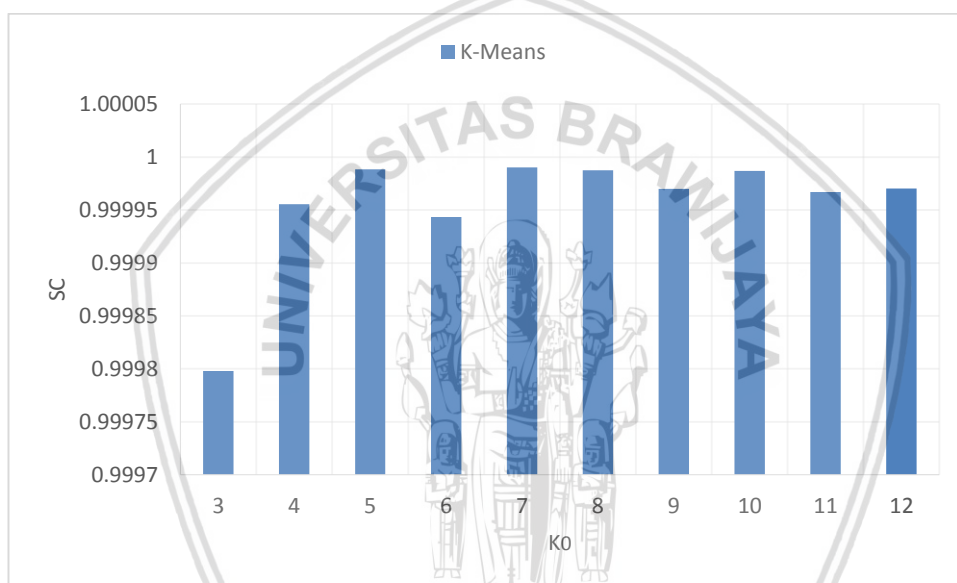


LAMPIRAN C UJI COBA ISODATA

C.1 Uji coba K_0

No	K_0	N_{min}	Max Var	d_{min}	Max Iteration	data	p1	p2	p3	p4	p5	rata- rata
1	3	100	0.9	0.5	50	10000	0.9997983	0.999799	0.999797	0.999798	0.999799	0.999798092
2	4						0.9997985	0.999995	0.999992	0.999996	0.999996	0.999955462
3	5						0.9999935	0.999988	0.999994	0.999984	0.999982	0.999988467
4	6						0.9999846	0.999984	0.999971	0.999978	0.999799	0.999943448
5	7						0.9999844	0.999988	0.999993	0.99999	0.999995	0.99999024
6	8						0.9999978	0.999982	0.999984	0.99998	0.999993	0.999987507
7	9						0.9999785	0.999993	0.999975	0.999935	0.999969	0.999970061
8	10						0.9999591	0.999995	0.999995	0.999997	0.999988	0.999986904
9	11						0.9999935	0.999982	0.999931	0.99996	0.999967	0.999966828
10	12						0.9999499	0.999971	0.999992	0.999995	0.999944	0.999970253
					Max							0.99999024

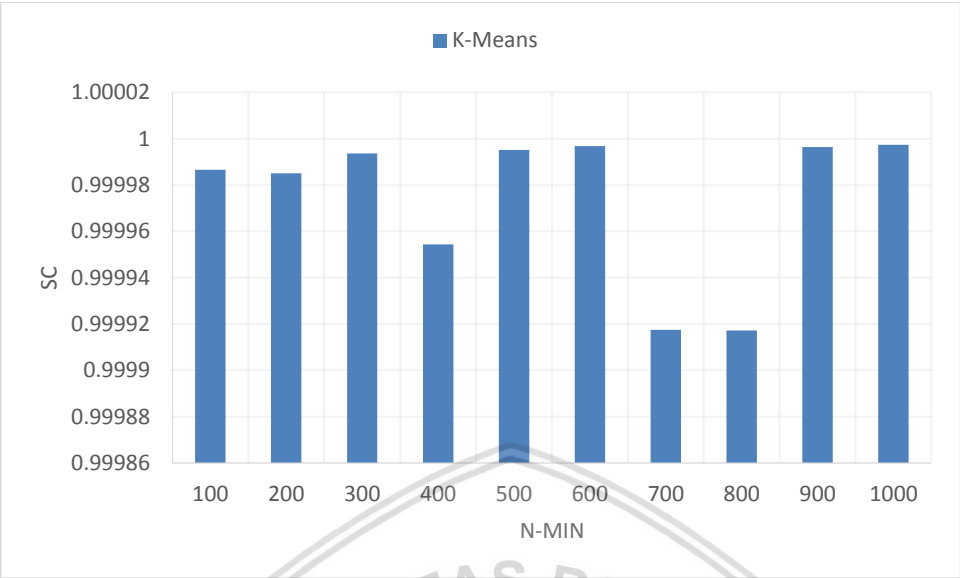
Grafik Uji Coba K_0



C.2 Uji coba n_{min}

No	K_0	N_{min}	Max Var	d_{min}	Max Iteration	data	p1	p2	p3	p4	p5	rata- rata
1	7	100	0.9	0.5	50	10000	0.9999941	0.999983	0.999991	0.999993	0.999972	0.9999866
2		200					0.9999939	0.999985	0.999992	0.999957	0.999997	0.999985043
3		300					0.9999972	0.999988	0.999993	0.999996	0.999993	0.999993602
4		400					0.9997984	0.999989	0.999997	0.999993	0.999995	0.999954407
5		500					0.9999948	0.999995	0.999993	0.999996	0.999997	0.999995114
6		600					0.9999959	0.999999	0.999995	0.999997	0.999997	0.99999678
7		700					0.9997984	0.999996	0.999997	0.999798	0.999997	0.999917432
8		800					0.9999953	0.999799	0.999998	0.999995	0.999799	0.999917173
9		900					0.9999953	0.999996	0.999998	0.999995	0.999998	0.999996399
10		1000					0.9999986	0.999997	0.999998	0.999996	0.999997	0.999997315
					Max							0.999997315

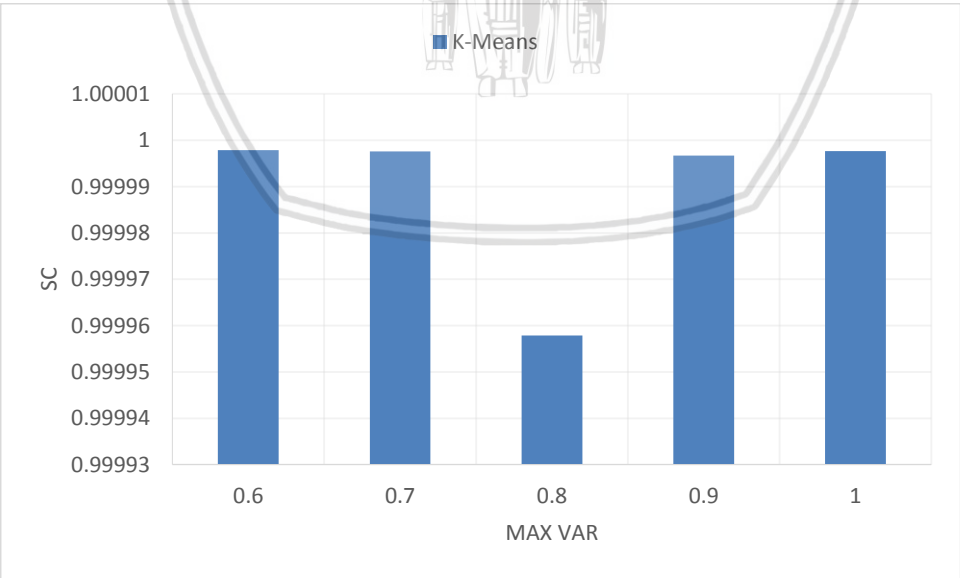
Grafik Uji coba n_{min}



C.3 Uji Coba variabel Varian max (σ^2_{max})

No	Ko	Nmin	Max Var	dmin	Max Iteration	data	p1	p2	p3	p4	p5	rata- rata
1	7	1000	0.6	0.3	50	10000	0.9999986	0.9999996	0.9999999	0.9999998	0.9999999	0.999997904
2			0.7				0.9999952	0.9999999	0.9999999	0.9999997	0.9999998	0.999997613
3			0.8				0.999798	0.9999996	0.9999999	0.9999999	0.9999998	0.999957847
4			0.9				0.9999971	0.9999995	0.9999997	0.9999997	0.9999996	0.999996672
5			1				0.9999984	0.9999999	0.9999997	0.9999997	0.9999998	0.999997677
Max												0.999997904

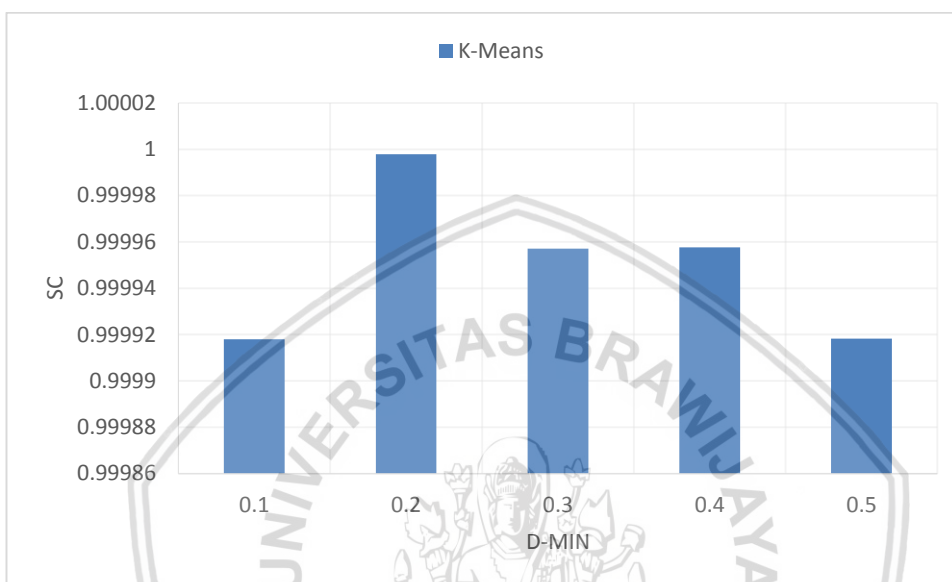
Grafik Uji coba Variabel max



C.4 Uji coba variabel d_{min}

No	Ko	Nmin	Max Var	dmin	Max Iteration	data	p1	p2	p3	p4	p5	rata-rata
1	7	1000	0.6	0.1	50	10000	0.9999986	0.999798	0.999799	0.999997	0.999998	0.999917989
2				0.2			0.9999971	0.999998	0.999999	0.999998	0.999998	0.999997913
3				0.3			0.9997985	0.999996	0.999997	0.999997	0.999997	0.999957133
4				0.4			0.9999982	0.999998	0.999799	0.999997	0.999997	0.99995766
5				0.5			0.9999972	0.999798	0.999998	0.999999	0.999799	0.99991824
Max												0.999997913

Grafik uji coba variabel d_{min}



C.5 Uji coba Iterasi max

No	Ko	Nmin	Max Var	dmin	Max Iteration	data	p1	p2	p3	p4	p5	rata-rata
1	7	1000	0.6	0.2	10	10000	0.9999988	0.999996	0.999997	0.999996	0.999999	0.999997452
2					20		0.9999961	0.999996	0.999996	0.999997	0.999999	0.999996696
3					30		0.9999972	0.999998	0.999998	0.999997	0.999997	0.999997601
4					40		0.9997985	0.999997	0.999799	0.999997	0.999799	0.999878193
5					50		0.9999985	0.999999	0.999799	0.999799	0.999999	0.999918816
6					60		0.9999985	0.999799	0.999799	0.999999	0.999997	0.999918235
7					70		0.9999972	0.999995	0.999996	0.999995	0.999997	0.999996067
8					80		0.9997984	0.999997	0.999799	0.999799	0.999997	0.999878124
9					90		0.9999964	0.999798	0.999997	0.999799	0.999997	0.999917482
10					100		0.9999976	0.999998	0.999997	0.999799	0.999997	0.999957662
Max												0.999997601

Grafik uji coba Iterasi Max

